MEMORANDUM
RM-5759-PR
OCTOBER 1969

TACTICS: A THREE-BODY, THREE-DIMENSIONAL INTERCEPT SIMULATION PROGRAM

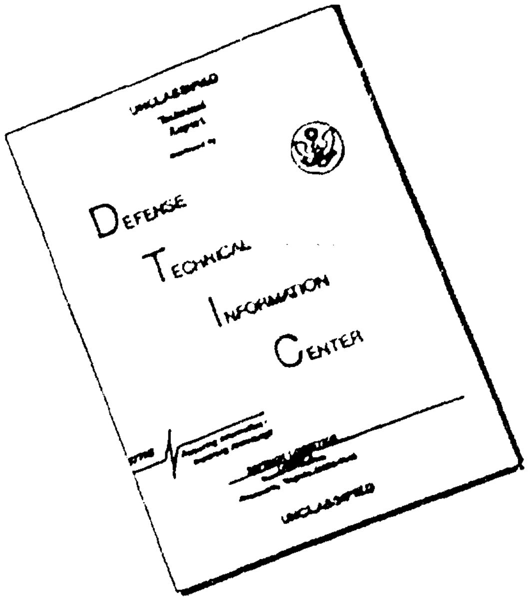
J. H. Hutcheson and R. L. Segerblom

PREPARED FOR:
UNITED STATES AIR FORCE PROJECT RAND

The RAND Corporation
SANTA MONICA • CALIFORNIA

186

DISCLAIMER NOTICE



THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

MEMORANDUM
RM-5759-PR
OCTOBER 1969

TACTICS: A THREE-BODY, THREE-DIMENSIONAL INTERCEPT SIMULATION PROGRAM

J. H. Hutcheson and R. L. Segerblom

This research is supported by the United States Air Force under Project RAND—Contract No. F44620-67-C-0045—monitored by the Directorate of Operational Requirements and Development Plans, Deputy Chief of Staff, Research and Development, Hq USAF. Views or conclusions contained in this study should not be interpreted as representing the official opinion or policy of the United States Air Force.

DISTRIBUTION STATEMENT
This document has been approved for public release and sale; its distribution is unlimited.

The RAND Corporation
1700 MAIN ST. • SANTA MONICA • CALIFORNIA • 90406

PREFACE

This Memorandum contains descriptive material and reference information necessary to understand and use TACTICS, a computer program which mathematically simulates the flight trajectories of as many as three different vehicles simultaneously. The program has been in use at Rand and elsewhere for about a year and a half, principally in studies of aircraft and missile performance in air-to-air combat and in air-to-surface missile (ASM) and surface-to-air missile (SAM) simulations. Because the simulation model is primarily useful as a research tool for studying interceptor-target guidance and interceptor trajectories in general, emphasis has been placed on providing versatility and flexibility for solving a wide variety of problems.

Magnetic tape copies of the program have been sent upon request to various facilities of the USAF and USN and contractors engaged in research in related fields, as well as to the armed forces (or affiliated institutions) of Japan, Canada, and Germany. The descriptive material should be of interest to those concerned with related fields of effort, while the reference information is necessary for a thorough understanding and effective use of the model.

SUMMARY

TACTICS is a computer program written in FORTRAN IV which mathematically simulates the dynamics of flight in three-dimensional space of as many as three vehicles simultaneously. The purpose of this Memorandum is to acquaint prospective users with the capabilities and basic theory of the program and to serve as a reference manual for those who wish to use the program.

The first part of the Memorandum contains the description and theory of operation and is oriented toward those with a mathematical or technical background. The second part is concerned with how to use the program, i.e., how to provide input data, select options, and develop a flight program. Wherever possible, FORTRAN symbols are relegated to this second part. A number of illustrative examples of a wide variety of problems are given in detail, including data and FORTRAN listings, in order to facilitate the use of TACTICS and the obtaining of results without detailed knowledge of the inner workings of the program. In many cases it should be possible to set up a specialized problem by modifying or combining various features of the examples.

ACKNOWLEDGMENTS

In developing TACTICS, the authors were fortunate in being able to draw upon the many varied experiences of the Rand staff. Since necessity is the mother of invention, credit for the initial concept is due to those who defined the needs so clearly and later contributed many ideas and suggestions leading to improvements. Although there were many, those principally involved were B. Boehm, T. F. Burke, T. B. Garber, T. E. Greene, J. Huntzicker, and D. N. Morris. Others responsible for translating ideas or concepts into computer language, inventing sample problems, cross-checking results, and debugging include J. Bedell, C. Fleming, J. Jolissaint, N. Maguire, and M. Samaniego. L. G. Martin and R. Spicer deserve special credit for many constructive suggestions and for inventing problem runs that were particularly effective in leading to improvements.

CONTENTS

FIGURES

SYMBOLS

FORTRAN Notation	Symbol*	Definition
A(I,J)	$\bar{1}_A$	Unit vector normal to velocity vector $\bar{V}$ and also in the horizontal plane
ABØØST(I)	a_B	Boost acceleration, assumed to be an average value (ft/sec^2)
ACØM(I)	a_C	Absolute magnitude of vehicle lateral acceleration (ft/sec^2, printout in g's)
ACØMA(I)	a_{Ch}	Horizontal component of lateral acceleration (g's)
ACØMD(I)	a_{CV}	Vertical component of lateral acceleration of vehicle(i) (g's)
ALPHA(I)	α	Angle of attack (deg)
ALPHAO(I)	α_o	Zero-lift angle of attack (deg)
ALT(I)	h	Altitude of the vehicle (ft)
AØUT(I)	a_o	Absolute magnitude of output lateral acceleration of vehicle (i) (g's)
AØUTA(I)	a_{ch}	Horizontal component of output lateral acceleration of vehicle (i) (g's)
AØUTD(I)	a_{oV}	Vertical component of output lateral acceleration of vehicle (i) (g's)
AREA(I)	A	Reference area (ft^2)
ASMAX(I)	a_{Smax}	Structural lateral acceleration limit (g's)
AZMAX(I)	η_{max}	Maximum azimuth angle (gimbal limit) (deg)
AZMUTH(1)	η_{12}	Azimuth angle in aircraft coordinates, vehicle 2 with respect to vehicle 1 (deg, + for right, - for left)
A1(I,J)	$\bar{1}_{A1}$	Unit vector normal to the axis of the aircraft and also in the horizontal plane
BANK(I)	ψ_B	Aircraft bank angle defined in relation to wind axis system (see Figs. 7 and 8) (deg)
BCØN(I)	$dC_D/d(C_L^2)$	Coefficient used with parabolic approximation for drag coefficient as a function of lift coefficient

*Subscripts "i" which indicate vehicle numbers have been omitted throughout the list for notational convenience.

FORTRAN Notation	Symbol	Definition
BEARNG(1)	B_{12}	Bearing angle in aircraft coordinates, vehicle 2 with respect to vehicle 1 (deg)
BEARNG(2)	B_{13}	Bearing angle in aircraft coordinates, vehicle 3 with respect to vehicle 1 (deg)
BEARNG(3)	B_{23}	Bearing angle in aircraft coordinates, vehicle 3 with respect to vehicle 2 (deg)
BEARNG(4)	B_{21}	Bearing angle in aircraft coordinates, vehicle 1 with respect to vehicle 2 (deg)
BEARNG(5)	B_{31}	Bearing angle in aircraft coordinates, vehicle 1 with respect to vehicle 3 (deg)
BEARNG(6)	B_{32}	Bearing angle in aircraft coordinates, vehicle 2 with respect to vehicle 3 (deg)
BETA(I)	β	Ballistic coefficient (lb/ft^2)
CDOCØN(I)	C_{D_o}	Zero-lift drag coefficient
CLMAX(I)	C_{Lmax}	Maximum aerodynamic lift coefficient
CØDRAG(I)	C_D	Aerodynamic drag coefficient
CØLIFT(I)	C_L	Aerodynamic lift coefficient
CWDØT(I)	$\dot{W}$	Time rate of change of weight (lb/sec)
D(I,J)	$\overline{1}_D$	Unit vector normal to the velocity vector and normal to vector $\overline{1}_A$, forming a right-hand system
DATA(200)		Initial-condition input data ranging from 1 to 200 (so far, only 143 are used)
DELV	ΔV	Boost velocity of missile (ft/sec)
DENS(I)	ρ	Air density (slug/ft^3)
DRAG(I)	D	Drag force (lb)
DTMIN		Minimum value for integration step size, used only for determination of miss distance (sec)
DTØ		Starting value for integration step size (sec)
DTPØ		Time interval for printing output (sec)
DVPHI(I)	$\Delta\gamma$	Assumed error in γ for aiming (deg)
DVTH(I)	$\Delta\theta$	Assumed error in θ_V for aiming (deg)
D1(I,J)	$\overline{1}_{D1}$	Unit vector normal to the axis of the aircraft and also in a plane normal to the horizontal plane

FORTRAN Notation	Symbol	Definition
ELEV(1)	ε_{12}	Elevation angle in aircraft coordinates, vehicle 2 with respect to vehicle 1 (deg, + for above, - for below)
ELEV(2)	ε_{13}	Elevation angle in aircraft coordinates, vehicle 3 with respect to vehicle 1 (deg, + for above, - for below)
ELEV(3)	ε_{23}	Elevation angle in aircraft coordinates, vehicle 3 with respect to vehicle 2 (deg, + for above, - for below)
ELEV(4)	ε_{21}	Elevation angle in aircraft coordinates, vehicle 1 with respect to vehicle 2 (deg, + for above, - for below)
ELEV(5)	ε_{31}	Elevation angle in aircraft coordinates, vehicle 1 with respect to vehicle 3 (deg, + for above, - for below)
ELEV(6)	ε_{32}	Elevation angle in aircraft coordinates, vehicle 2 with respect to vehicle 3 (deg, + for above, - for below)
ELEVMX(1)	ε_{max}	Maximum elevation angle (gimbal limits) (deg)
EXTR(20)		Extra quantities (real) in CØMMØN for optional use (the first six are part of standard printout)
G	g	Mass conversion (32.174 ft/sec^2); also used as constant gravitational attraction for flat-earth option
GFØRC		Total lateral acceleration, including gravitational effects specified for a maneuver (g's)
GFØRCE(I)	F_n	Total lateral acceleration, including gravitational effects (g's)
HM		Actual integration step size used by the program (sec)
HMIN		Minimum specified value for integration step size used in variable Adams-Moulton mode integration (sec)
HMX		Maximum specified value for integration step size used in variable Adams-Moulton mode integration (sec)
IAERØ		Flag indicating aerodynamic option
IEXTRA(10)		Ten extra integer quantities for optional use (CØMMØN package)

FORTRAN Notation	Symbol	Definition
ILAUN		Flag indicating sequence of steps involved in missile launch (see Appendix H)
IMISS		Flag indicating that the program should continue after finding point of closest approach (miss distance)
IMPLSE(I)	I	Specific impulse of rocket motor (sec)
INERF, INERT		Flag indicating that a vehicle's velocity is in relation to a rotating or non-rotating (inertial) coordinate system
EPSLØN	ϵ	Threshold value used in on-off control laws for stability
ERTEST		Maximum allowable relative truncation error for Adams-Moulton variable step size integration mode
IPRINT(20)		Flag indicating printout option
IRØT8		Flag indicating option for rotating or nonrotating earth
ISTØRE		Flag indicating that position and velocity values are to be stored at the time of launch
ITAU(I)		Flag indicating number of vehicle first-order time lags
ITHR		Flag indicating thrust option
JATMØS		Flag specifying whether initial-condition value of velocity is expressed in Mach number or ft/sec
JINTEG		Flag specifying integration mode
JPØL, KPØL, LPØL, MPØL, NPØL		Flags used in PØLICY subroutine
JVEH(I)		Flag indicating aerodynamic option for tables
KINTEG		Flag indicating whether a round-earth or flat-earth option is to be used
KLAUN		Decimal fraction of missile's maximum range at which it is to be launched
LAMDAO(I)	λ_0	Navigation constant for closed-loop guidance routines
LAT(I)	ϕ	Geocentric latitude of the vehicle (deg)
LATO	ϕ_0	Latitude of the local coordinate system origin (deg)

FORTRAN Notation	Symbol	Definition
LEVEL(I)		Flag set in subroutine STRLVL used to communicate to PØLICY that the vehicle velocity vector is in a horizontal plane within a tolerance of 0.002 rad
LIFT(I)	L	Aerodynamic lift force (lb)
LØNG(I)	Λ	Longitude of the vehicle (deg)
LØNGO	Λ_o	Longitude of the origin of the local coordinate system (deg)
MACH(I)	M	Mach number
MACHMX(I)	M_{max}	Placard limit (maximum Mach number for vehicle)
MINMR		Missile range to target within which program will automatically initiate process for miss distance computation (ft)
MØDE		Flag indicating captive flight option
NPRINT		Flag indicating number of sections of output to be printed
ØMEGA(I,J)	$\bar{\omega}$	Angular rate vector of vehicle (i) with respect to x, y, z coordinate frame (rad/sec)
ØMEGAB(1,J)	ω_{B12}	Angular rate bias term used for predictive guidance, vehicle 2 with respect to vehicle 1 (rad/sec)
ØMEGAB(2,J)	ω_{B13}	Angular rate bias term used for predictive guidance, vehicle 3 with respect to vehicle 1 (rad/sec)
ØMEGAB(3,J)	ω_{B23}	Angular rate bias term used for predictive guidance, vehicle 3 with respect to vehicle 2 (rad/sec)
ØMEGAB(4,J)	ω_{B21}	Angular rate bias term used for predictive guidance, vehicle 1 with respect to vehicle 2 (rad/sec)
ØMEGAB(5,J)	ω_{B31}	Angular rate bias term used for predictive guidance, vehicle 1 with respect to vehicle 3 (rad/sec)
ØMEGAB(6,J)	ω_{B32}	Angular rate bias term used for predictive guidance, vehicle 2 with respect to vehicle 3 (rad/sec)
ØMEGAE	ω_e	Earth's angular rate of rotation ($7.29211585 \times 10^{-5}$ rad/sec)

FORTRAN Notation	Symbol	Definition
ØMEGAR(1,J)	ω_{R12}	Relative angular rate vector, vehicle 2 with respect to vehicle 1 (rad/sec)
ØMEGAR(2,J)	ω_{R13}	Relative angular rate vector, vehicle 3 with respect to vehicle 1 (rad/sec)
ØMEGAR(3,J)	ω_{R23}	Relative angular rate vector, vehicle 3 with respect to vehicle 2 (rad/sec)
ØMEGAR(4,J)	ω_{R21}	Relative angular rate vector, vehicle 1 with respect to vehicle 2 (rad/sec)
ØMEGAR(5,J)	ω_{R31}	Relative angular rate vector, vehicle 1 with respect to vehicle 3 (rad/sec)
ØMEGAR(6,J)	ω_{R32}	Relative angular rate vector, vehicle 2 with respect to vehicle 3 (rad/sec)
PHIDØT(I)	$\dot{\varphi}$	Time rate of change of φ (rad/sec)
PRES(I)	p	Air pressure (lb/ft^2)
Q(I)	q	Dynamic pressure (lb/ft^2)
R(I,J)	$\bar{r}$	Range vector from x, y, z origin to vehicle (i) (ft)
RAD	$\pi/180^\circ$	Factor for converting degrees to radians (rad/deg)
RDØT(1)	$\dot{r}_{12}$	Range rate, vehicle 2 with respect to vehicle 1 (ft/sec)
RDØT(2)	$\dot{r}_{13}$	Range rate, vehicle 3 with respect to vehicle 1 (ft/sec)
RDØT(3)	$\dot{r}_{23}$	Range rate, vehicle 3 with respect to vehicle 2 (ft/sec)
RDØT(4)	$\dot{r}_{21}$	Range rate, vehicle 1 with respect to vehicle 2 (ft/sec)
RDØT(5)	$\dot{r}_{31}$	Range rate, vehicle 1 with respect to vehicle 3 (ft/sec)
RDØT(6)	$\dot{r}_{32}$	Range rate, vehicle 2 with respect to vehicle 3 (ft/sec)
RLAUN		Range at which missile is to be launched, used in LEADCL subroutine (ft)
RMTMAX		Maximum range of missile (ft)
RØLL(I)	ψ	Aircraft roll angle (deg) defined in relation to aircraft axes coordinate system
RØLLL		Total angle through which vehicle is to roll (deg)

FORTRAN Notation	Symbol	Definition
RØLLR8(I)	$\dot{\psi}$	Time rate of change of aircraft roll angle (deg/sec)
RR(I,J)	$\overline{R}$	Geocentric range vector (ft) directed radially from the earth's center
RRDØT(I,J)	$\dot{\overline{R}}$	Velocity of the topocentric range vector with respect to inertial space (ft/sec)
RREL(1,J)	$\overline{r}_{12}$	Relative range vector, vehicle 2 relative to vehicle 1 (ft)
RREL(2,J)	r_{13}	Relative range vector, vehicle 3 relative to vehicle 1 (ft)
RREL(3,J)	r_{23}	Relative range vector, vehicle 3 relative to vehicle 2 (ft)
RREL(4,J)	r_{21}	Relative range vector, vehicle 1 relative to vehicle 2 (ft)
RREL(5,J)	r_{31}	Relative range vector, vehicle 1 relative to vehicle 3 (ft)
RREL(6,J)	r_{32}	Relative range vector, vehicle 2 relative to vehicle 3 (ft)
RO	R_0	Average radius of the earth (20,902,287 ft)
RO(I)	$\overline{R}_0$	Position of the local x, y, z coordinate system origin with respect to inertial space (ft)
RODØT(I)	$\dot{\overline{R}}_0$	Velocity of the local x, y, z coordinate system with respect to inertial space (ft/sec)
SGAMA(I)	σ	Absolute magnitude of the angle between the velocity vector of the interceptor and the line of sight from interceptor to target (rad)
SLØPE(I)	$dC_L/d\alpha$	Slope of C_L versus α curve
SØUND(I)	s	Speed of sound (ft/sec)
TABCØN(I)	T_{ab}	Constants to be used for afterburner thrust (lb)
TAU(I,J)	τ	First-order approximation for overall missile response time
TBURN1		First-stage burning time (sec)
TBURN2		Second-stage burning time (sec)
TEMP(I)	T.	Temperature (degrees Fahrenheit)

FORTRAN Notation	Symbol	Definition
TGUIDE(I)		Time interval missile is to fly unguided after launch (sec)
THCØN(I)	T_M	Constants to be used for military thrust (lb)
THDØT(I)	$\dot{\theta}$	Time rate of change of theta (rad/sec)
THRØTL(I)	K	Constant used to multiply thrust to represent throttle setting
THRUST(I)	T	Propulsive thrust force (lb)
TIME	t	Running time (sec)
TLAUN(I)	t_L	Launch time (sec)
TØTAL		Time value at which program is to stop
UNITL(I,J)	$\bar{1}_L$	Unit vector directed along the lift vector
UNITLL(I,J)	$\bar{1}_{LL}$	Unit vector directed eastward along the local parallel of latitude
UNITPP(I,J)	$\bar{1}_P$	Unit vector directed northward along the local meridian of longitude
UNITPV(I,J)	$\bar{1}_\perp$	Unit vector normal to velocity vector $\bar{V}$ and along the net lateral acceleration vector
UNITR(K,J)	$\bar{1}_r$	Unit vector directed along the relative range vectors and using the same subscript notation
UNITRR(I,J)	$\bar{1}_R$	Unit vector directed radially from the earth's center along the vector $\bar{R}$ (RR(I,J))
UNITT(I,J)	$\bar{1}_T$	Unit vector directed along the longitudinal axis of the vehicle (i) and also assumed to be coincident with the thrust vector $\bar{T}$
UNITV(I,J)	$\bar{1}_V$	Unit vector corrected along velocity vector $\bar{V}$
V(I,J)	$\bar{V}$	Velocity vector (ft/sec)
VDØT(I)	$\dot{V}$	Rate of acceleration or deceleration along the trajectory, i.e., changes in speed (ft/sec^2)
VREL(1,J)	$\bar{V}_{12}$	Relative velocity, vehicle 2 with respect to vehicle 1 (ft/sec)
VREL(2,J)	$\bar{V}_{13}$	Relative velocity, vehicle 3 with respect to vehicle 1 (ft/sec)

FORTRAN Notation	Symbol	Definition
VREL(3,J)	$\bar{V}_{23}$	Relative velocity, vehicle 3 with respect to vehicle 2 (ft/sec)
VREL(4,J)	$\bar{V}_{21}$	Relative velocity, vehicle 1 with respect to vehicle 2 (ft/sec)
VREL(5,J)	$\bar{V}_{31}$	Relative velocity, vehicle 1 with respect to vehicle 3 (ft/sec)
VREL(6,J)	$\bar{V}_{32}$	Relative velocity, vehicle 2 with respect to vehicle 3 (ft/sec)
WBURN(I)	W_B	Weight of missile at burnout (lb)
WDØT(I)	$\dot{W}$	Time rate of change of weight (lb/sec)
WEIGHT(I)	W	Weight (lb)
WO(I)	W_o	Initial weight of vehicle (i) (lb)

Part 1

DESCRIPTION AND THEORY OF OPERATION

I. INTRODUCTION

TACTICS is a computer program written in FORTRAN for use in simulating the kinematics and dynamics of motion of three vehicles in three-dimensional space. It was developed primarily as a research tool for use in detailed explorations of the mechanics, geometry, and vehicle performance characteristics of an interceptor-target engagement. The output is a step-by-step time history of variables relating to each of three vehicles' position, velocity, acceleration, applied forces, attitude or orientation, and aerodynamics. While the program is highly versatile, its most important capabilities relate to interceptor-target guidance and intercept trajectories in general. Since the flights of three vehicles may be represented simultaneously, the program can be used to simulate aerial combat between aircraft (e.g., a two-on-one engagement, or a one-on-one with missile launch). There are also a number of other possibilities, such as (1) using one or more of the vehicles to represent an air-to-surface missile (ASM), surface-to-air missile (SAM), or surface-to-surface missile (SSM), (2) using the target to represent an ICBM reentry vehicle (RV), (3) using vehicles 1 and 2 to represent first- and second-stage boosters, or (4) using one vehicle to represent an orbiting satellite.

So far, the model has been used primarily in simulating fighter-versus-fighter and missile-versus-fighter duels. Because of the number of maneuver routines created for this specialized purpose, the model's present development in this area is farther advanced than in other areas. However, ASM and SAM simulations are currently being performed in connection with other Rand projects, and the program has been used successfully in satellite-intercept problems. The model's usefulness and potential capabilities are expected to grow with each new application.

In the organization of TACTICS, primary emphasis was placed on versatility in order to accommodate a broad spectrum of problems. At the same time, it was thought that the program should be easy to use and easily adaptable to refinements and new features. Accordingly, it was built in modular building-block form, with many subroutines that

ca be replaced, modified, or simply disregarded at the option of the user. In fact, new building blocks are welcomed, since they increase the program's potential problem-solving capability. The construction of TACTICS is in many respects comparable to that of the ROCKET[1] program, although the two have different purposes. Many ideas were borrowed from ROCKET in the formulation of TACTICS, especially its most important distinguishing feature, that of allowing the researcher to obtain results without a detailed knowledge of the inner workings of the program.

The program is able to simulate the flight or trajectory of almost any type of vehicle in almost any mixture of vehicles. Various guidance-law subroutines are available for simulating terminal-homing or command-guided trajectories (e.g., biased proportional and proportional navigation, lead collision, and pursuit and lead pursuit). A variety of open-ended control-law subroutines are also available for simulating aerodynamic maneuvers such as turning, diving, climbing, and combinations thereof. The library of guidance- and control-law subroutines has been growing and will probably continue to grow as new problems are encountered. In fact, experience has shown that it is convenient to incorporate into one of these control-law subroutines the unique features of a particular vehicle or problem (e.g., Sidewinder or a hypothetical aircraft or missile design).

Options are available for considering the earth either flat or round and either fixed or rotating, but the gravitational field is a simple inverse square law field. Each of the three vehicles has three wind axes associated with its attitude or orientation reference system. The vehicles may be either fixed or in motion with respect to the earth but not below its surface. The miss distance or point of closest approach between two of three vehicles may be calculated and the problem run automatically terminated if so desired.

As mentioned earlier, the step-by-step time history of the engagement is limited to the consideration of the performance of three vehicles at one time. However, two devices have been incorporated for extending this capability to more than three vehicles by means of sequential computations. These devices, designated "recall" and "restore," are

illustrated in Fig. 1. Consider a vehicle 1 which launches a missile 2 at a target 3. At some subsequent time or event (e.g., a miss) it may be desirable to recall vehicle 2 and place it in captive flight on either 1 or 3 for a subsequent quasi-vehicle 4. (It is not necessary that 4 have the same characteristics as, or even resemble, 2.) To illustrate the restore feature, consider the same example except that at some time or event subsequent to the launching of vehicle 2 (e.g., a hit, miss, or ground impact) we wish to restore the launch-time situation. After restoration has occurred, events may proceed, and a new launch (4, 5, etc.) may take place at a subsequent time or event. Or perhaps branching is desired, i.e., some characteristic or parameter is altered and 4, 5, etc. are to be launched under the same restored conditions of time and geometry.

Consider the factors vital to any program for simulating flight trajectories. As illustrated in Fig. 2, it is necessary to (1) read in data as initial conditions, (2) calculate the geometry, (3) specify the applied forces involved, (4) integrate the equations of motion, and (5) output the answers. This is the basic framework of TACTICS, to which embellishments are added (e.g., model atmosphere, coordinate transformations, and aerodynamic computations). Although several options are provided for modes of integration and output form, the framework may be considered inflexible except for specifying the applied forces. In this case, maximum flexibility is provided by an arrangement comparable to plug-in modules; that is, the applied forces and the time(s), event(s), or situation(s) dictating their application are contained within two subroutines which may be specialized to deal with a particular problem. To illustrate, each particular problem is defined by initial-condition input data and by a number of PØLICY statements, which are logical expressions dictating the control laws governing the flight of each vehicle. They are usually conditionally based on time or on geometric and kinematic relationships. (The PØLICY subroutine is discussed in more detail in Section II.)

A rudimentary description of the operational principles necessary to simulate an intercept problem will now be useful. As a starting point, imagine a vehicle in three-dimensional space having a position

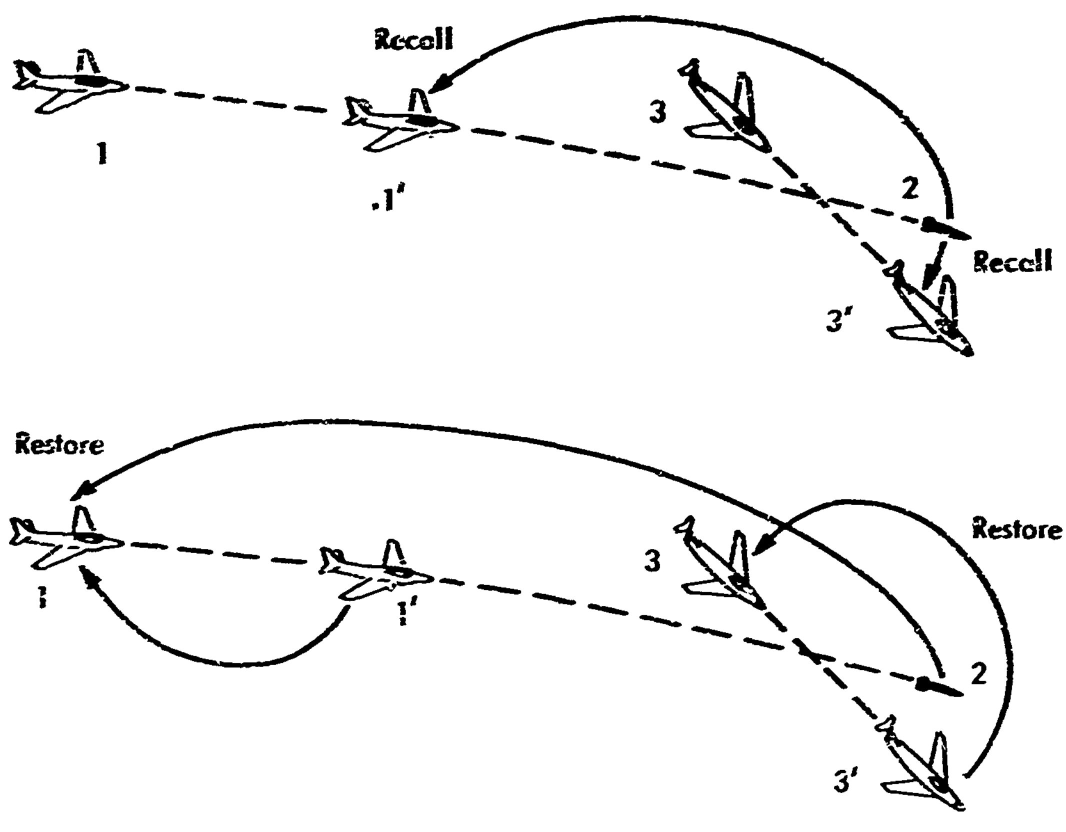


Fig. 1 — Recall and restore features contrasted

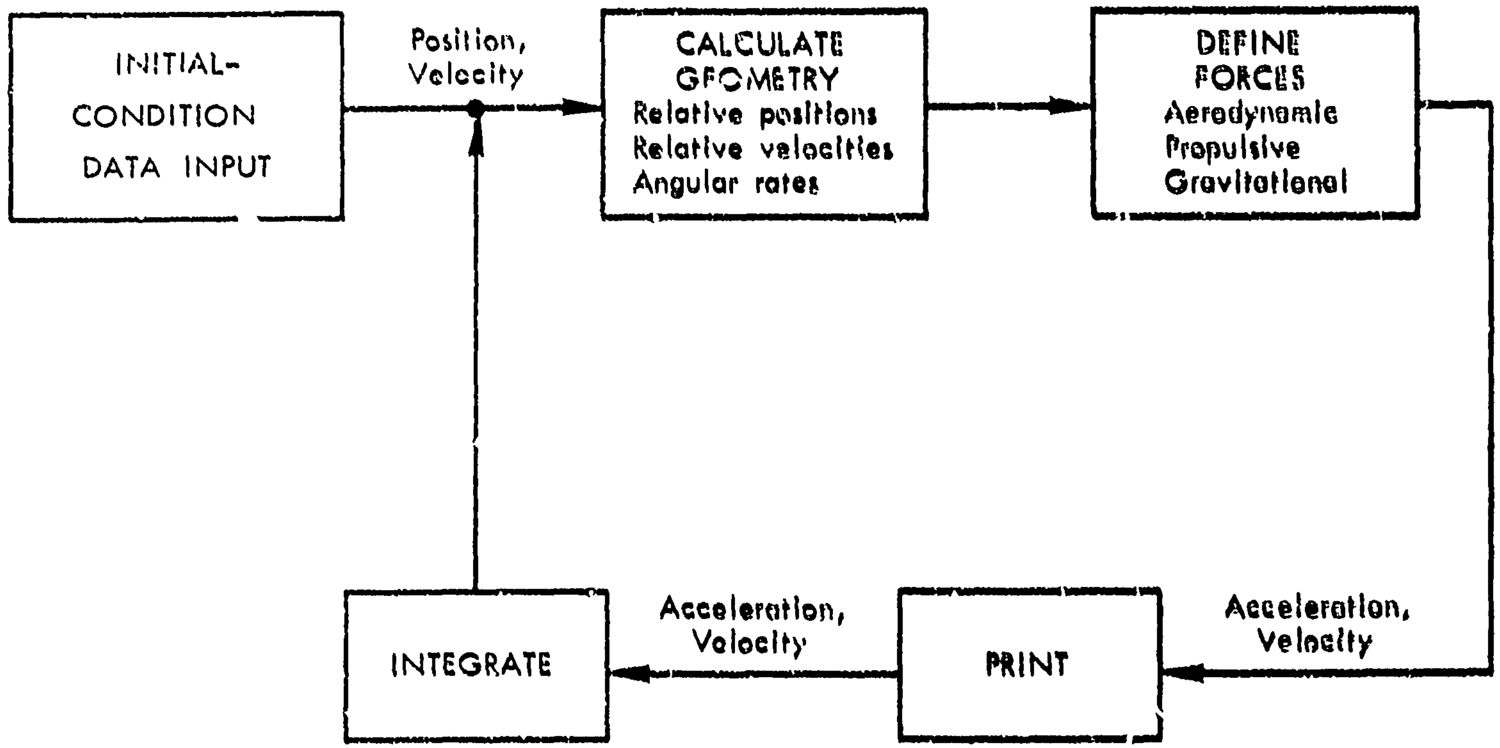


Fig. 2 — Basic framework for flight simulation (generalized)

and a velocity defined by initial-condition input data. If no accelerations or applied forces are involved, the time history or trajectory of the vehicle will obviously be a straight line. However, in the general trajectory case, there will be a net acceleration ($\ddot{x}$, $\ddot{y}$, $\ddot{z}$) due to gravitational, aerodynamic, and thrust forces, where the correspondence between the resultant force and $\ddot{x}$, $\ddot{y}$, $\ddot{z}$ is given by $F = ma$. If it is assumed that all forces can be defined and specified, the trajectory simulation problem is reduced to (1) integrating the net acceleration as a function of time to obtain velocity and (2) integrating velocity to obtain position. If the applied net force were constant or a simple function of time, the trajectory simulation would be straightforward and relatively simple. In the general case, particularly when closed-loop guidance and aerodynamics are involved, the forces may be complicated functions of position, velocity, geometry, and time (and may involve the behavior or predicted behavior of some other vehicle(s)); hence numerical integration techniques must be used to calculate the trajectory stepwise using time increment Δt. It should be clear from the preceding discussion that to start a problem run, input data for initial position and velocity must be supplied. The bulk of all other input data will pertain to parameters associated with the definition or calculation of the forces to be applied (e.g., lift, drag, thrust, and gravity). Representing the motion of three different vehicles multiplies the input requirements, of course, but not necessarily by a factor of three. The definition of the intercept problem, in terms of logic, timing, and geometric events, or of any one of these, dictates what types of control laws are to be applied under which circumstances. This definition is contained in a PØLICY subroutine that in a sense is also an input to the program. The forces or force functions which are to be called upon by PØLICY are contained in the library of control-law subroutines. It is not expected that this library will ever include every conceivable force function or guidance law, especially since TACTICS is a research tool for experimentation. However, experience has shown that new subroutines may usually be conveniently generated by modifying those already on hand.

II. PØLICY SUBROUTINE

The PØLICY portion of the program, which is supplied by the user, consists of logical statements and expressions dictating the guidance- or control-law subroutines that govern the flight of each vehicle. These subroutines define the various forces that are to be applied to a vehicle to perform a certain maneuver or to guide in accordance with some prescribed guidance law. The result of all forces will be a net acceleration. One need not be familiar with the mathematics involved in guidance, aerodynamics, propulsion, etc. to write a PØLICY routine. However, a large number of options are available, and the user should be familiar with them and with the library of available guidance- or control-law routines. To illustrate the use of a PØLICY subroutine, consider the problem of simulating the flight of an aircraft from take-off to landing. Clearly, a number of control laws defining the forces and hence the net acceleration would be needed for even the most elementary flight plan. This subject is discussed in further detail in Sections IV and IX. Consider the net accelerations associated with each of the following:

- o Climb (vertical plane).
- o Straight flight, i.e., no turning component of acceleration.
- o Turn (horizontal plane).
- o Dive (vertical plane).

Moreover, assume that subroutines are available for describing the above trajectories (CLIMB, STRFLT, RTURN, LTURN, and DIVE, respectively). A PØLICY routine is required to define the conditions dictating the transition from one maneuver to another. For example, CLIMB may be called for a given time interval or until some specified altitude or other criterion is reached; then STRFLT is called, and so on.

With respect to the preceding illustration, the most obvious example of choice of option is the selection of the CLIMB, STRFLT, etc. routines. In the argument listings for these routines there are also options pertaining to the following conditions:

- The vehicle the law is to govern (1, 2, or 3).
- The magnitude (and perhaps direction) of the propulsive or thrust force.
- The number of g's commanded for the climb, dive, or turn maneuvers.
- The use of tabulated values or analytic functions in calculating aerodynamic forces.

Appendix D contains a list of optional subroutines currently available and instructions for their use. Section IX gives examples of problem runs with corresponding PØLICY subroutines.

So far this subject has been discussed in the context of choices to be made within the PØLICY routine. Numerous other options may be selected by reading in flags or constants as part of initial-condition data. However, it is sometimes desirable to override these initial instructions in PØLICY if during a problem run a situation arises that requires, perhaps, a change in printout frequency or integration step size.

III. INPUT-OUTPUT GEOMETRY

In order to construct a problem run, it is necessary to define the problem by reading in initial-condition data and establishing a flight-control program (PØLICY subroutine). Before a detailed description of this process is possible, it is essential to have a basic acquaintance with certain fundamentals of the coordinate system and intercept geometry.

ABSOLUTE AND RELATIVE POSITION GEOMETRY

Figure 3 shows two position vectors, $\bar{r}_1$ and $\bar{r}_2$, in a space referred to a three-dimensional x, y, z coordinate system of arbitrary origin. The position of point 1 may be defined either in Cartesian coordinates, x_1, y_1, z_1, or in spherical coordinates, r_1, θ_1, φ_1. Because there are certain advantages to each form of coordinate system, TACTICS can convert one to the other, so that all vectors are expressed in six-element form, as in the following:

$$r_1 = r_1 \ (x, y, z, r, \theta, \varphi)$$

Initial-condition position data may be read in either Cartesian or spherical coordinates with respect to the reference frame. Generally, the former is the most convenient for this purpose. For example, it is easy to visualize that one vehicle is initially at an altitude of z_1 ft and arbitrarily placed at $x_1 = 0$, $y_1 = 0$, while perhaps another vehicle is initially at an altitude of z_2 ft and displaced horizontally from the first by x_2, y_2 ft.

Once initial conditions have been established and the vehicles located in a reference coordinate frame, it is generally true that the absolute displacements of the vehicles with respect to this frame are no longer of primary interest. Intercept problems are mainly functions of relative geometry, i.e., relative position and velocity or components thereof.

In Fig. 3, the vector $\bar{r}_{12}$ represents the range and direction of point 2 relative to point 1. For three vehicles, there is the vector $\bar{r}_3$, not shown. More generally, then, the relative position vectors are

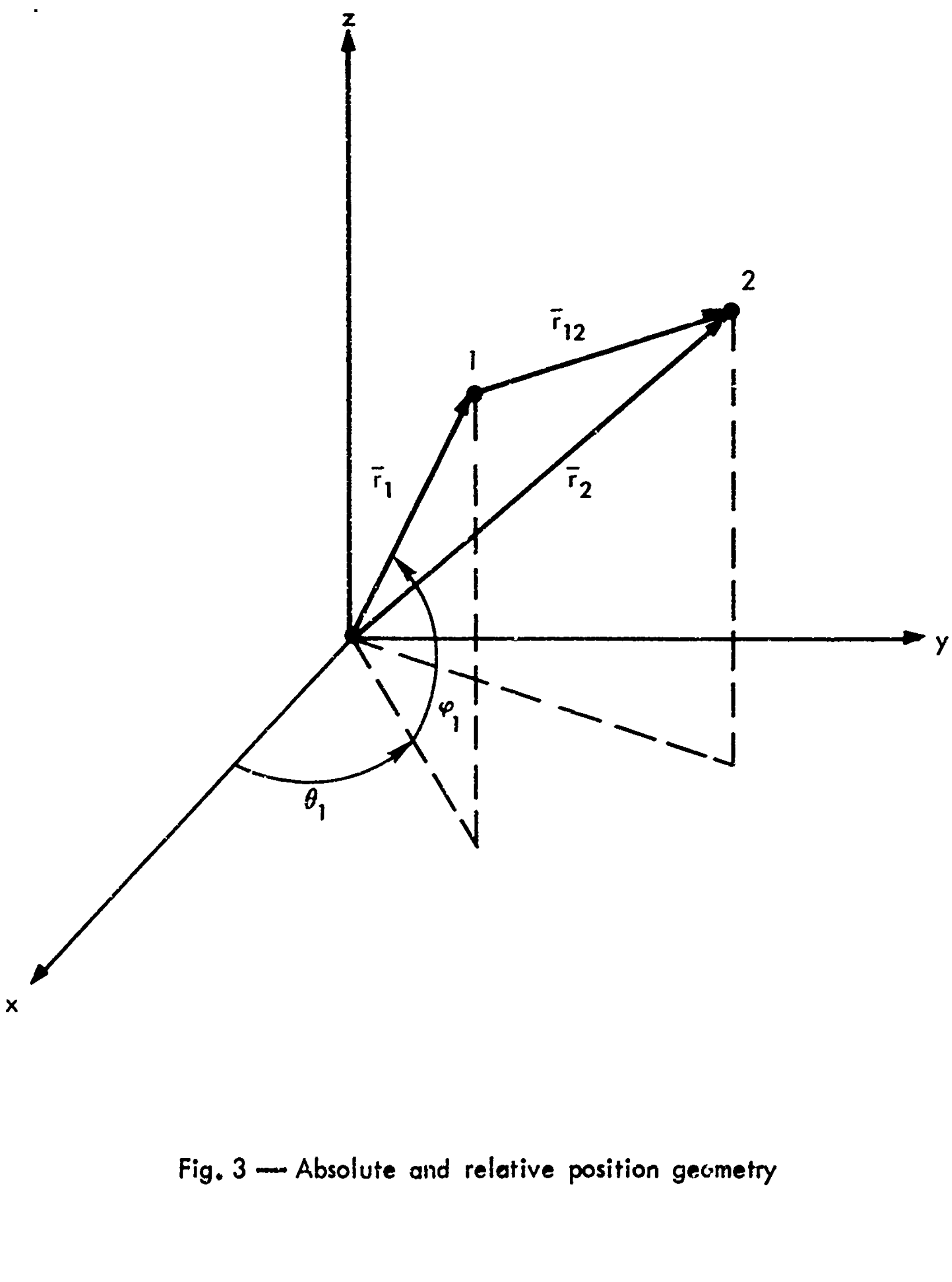


Fig. 3 — Absolute and relative position geometry

$$\begin{aligned} \overline{r}_{ij} &= \overline{r}_j - \overline{r}_i \\ \overline{r}_{ik} &= \overline{r}_k - \overline{r}_i \\ \overline{r}_{jk} &= \overline{r}_k - \overline{r}_j \end{aligned} \tag{1}$$

These quantities, as well as all other quantities pertaining to relative geometry (usually the entire intercept problem), are calculated and used extensively within the program and form an important portion of the printout. As an example, guided-missile control laws and criteria for missile launch are usually dependent on relative-position and angular-rate data. Because relative-position information is visualized and interpreted more conveniently in spherical coordinates, it is printed out in this form.

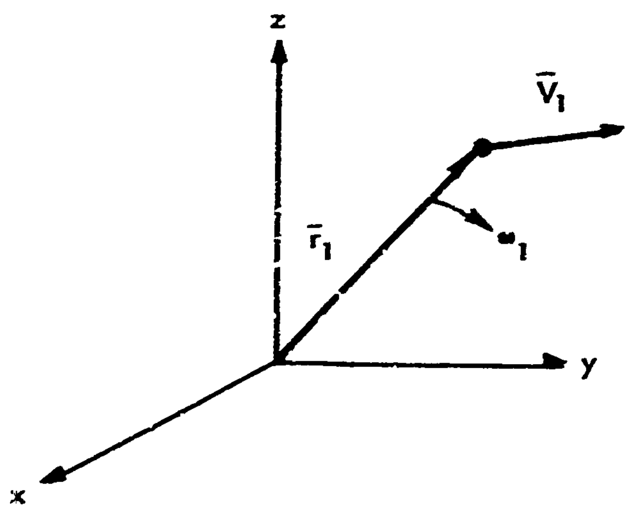
In summary:

- o Initial-condition position data may be specified in either Cartesian (x, y, z) or spherical (r, θ, φ) coordinates with respect to the reference frame. Units of measurement are feet and degrees.
- o All relative geometry (e.g., $\overline{r}_{12}$) is computed within the program and is printed in spherical coordinates.

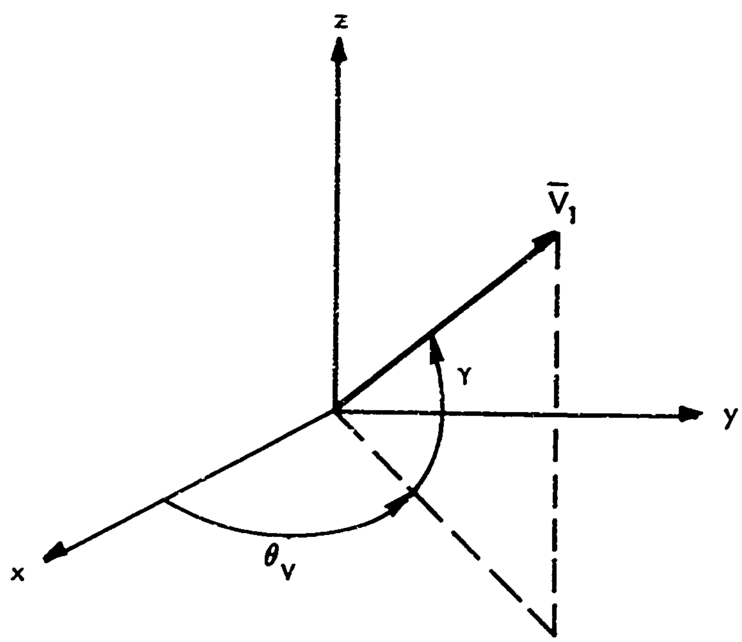
ABSOLUTE AND RELATIVE VELOCITY GEOMETRY

Input velocity data and position data are treated almost identically to one another. Figure 4(a) shows a velocity vector $\overline{V}_1$ associated with the position vector $\overline{r}_1$. For initial conditions, $\overline{V}_1$ may be entered in either Cartesian (V_x, V_y, V_z) or spherical (V, θ_V, γ) form (see Fig. 4(b)). Conversion from one form to the other takes place within the program. The unit of measurement is feet per second.

As in the preceding discussion of position data, the *relative* velocity vectors are calculated within the program and occupy significant portions of the printout. Similarly, the relative velocities are given by



(a) Absolute position and velocity



(b) Absolute velocity

Fig. 4 — Absolute geometry

$$\dot{\bar{r}}_{ij} = \dot{\bar{r}}_j - \dot{\bar{r}}_i \qquad \begin{array}{l} i = 1, 2, 3 \\ j = 1, 2, 3 \\ i \neq j \end{array} \tag{2}$$

and so on for $\dot{\bar{r}}_{jk}$, $\dot{\bar{r}}_{ik}$, etc. As with position vectors, coordinate transformations are performed so that these vectors are also expressed six-dimensionally in terms of $\dot{x}$, $\dot{y}$, $\dot{z}$, V, θ_V, and γ. It is frequently necessary in the computations to call upon both absolute ($\bar{r}_i$ and $\dot{\bar{r}}_i$) and relative ($\bar{r}_{ij}$ and $\dot{\bar{r}}_{ij}$) quantities. The angular rates of rotation of the range vectors, both absolute and relative, are of basic importance to most guidance or trajectory problems. Expressed in vector notation,

$$\dot{\bar{r}}_i = \dot{r}_i \bar{1}_{ri} + \bar{\omega}_i \times \bar{r}_i \tag{3}$$

where $\dot{r}_i$ = scalar time rate of change of the vector $\bar{r}_i$

$\bar{1}_{ri}$ = a unit vector along $r_i \left(\bar{1}_{ri} = \bar{r}_i/|r_i|\right)$

$\bar{\omega}_i$ = angular-rate vector orthogonal to both $\bar{r}_i$ and $\dot{\bar{r}}_i$

Similarly, for the relative velocity vectors

$$\dot{\bar{r}}_{ij} = \dot{r}_{ij} \bar{1}_{rij} + \bar{\omega}_{ij} \times \bar{r}_{ij} \tag{4}$$

It is important to note that for approaching vehicles the relative range rate $\dot{r}_{ij}$ is negative; for vehicles which are separating, it is positive. Also, the point of closest approach or minimum miss distance between two vehicles occurs when the absolute value of this relative range rate is zero.

Changing to spherical notation for convenience, let us consider the vector velocities $\dot{\bar{r}}_i$ and $\dot{\bar{r}}_j$ or $\bar{V}_i$ and $\bar{V}_j$, respectively. The following relationship for the angular-rate vector representing the rotation of the vector $\bar{r}_{ij}$ is applicable:

$$\bar{\omega}_{ij} = \bar{r}_{ij} \times \frac{\left(\bar{V}_j - \bar{V}_i\right)}{r_{ij}^2} \tag{5}$$

Note that with these basic relationships it is possible to resolve vector velocities into two useful components, one along $\overline{r}_{ij}$ and the other transverse to $\overline{r}_{ij}$. Also, the angular rotation of the line of sight (LOS) between $\overline{r}_i$ and $\overline{r}_j$ is known both in direction and magnitude. This latter quantity is of prime importance in most terminal (or command) missile or aircraft guidance applications. In actual practice, it is usually a quantity determined by rate gyro measurements (in missiles) or by processing θ, φ angle data measured by a ground radar.

So far, relationships have been given for resolving the velocity vectors into components parallel and perpendicular to the $\overline{r}_i$ and $\overline{r}_{ij}$ values and for calculating the angular-rate vectors. In subsequent discussions of guidance laws and acceleration, two useful vectors associated with velocity will become important. Imagine a plane normal to a velocity vector $\dot{\overline{r}}$ having spherical coordinates V, θ_V, γ. It is convenient and useful to define a unit vector $\overline{1}_A$ common to this plane and the horizontal x-y plane, and a unit vector $\overline{1}_D$ common to this plane and the vertical plane. Thus, $\dot{\overline{r}}$ or $\overline{V}$ and the unit vector $\overline{1}_V$ will form a right-hand orthogonal system with $\overline{1}_A$ and $\overline{1}_D$, as shown in Fig. 5. The following relationships are applicable for determining the components:

$$\begin{aligned}
1_{Ax} &= -\sin\theta_V & 1_{Dx} &= -\sin\gamma\cos\theta_V & 1_{Vx} &= \cos\gamma\cos\theta_V \\
1_{Ay} &= \cos\theta_V & 1_{Dy} &= -\sin\gamma\sin\theta_V & 1_{Vy} &= \cos\gamma\sin\theta_V \qquad (6) \\
1_{Az} &= 0 & 1_{Dz} &= \cos\gamma & 1_{Vz} &= \sin\gamma
\end{aligned}$$

ELEVATION, AZIMUTH, AND BEARING-ANGLE GEOMETRY

An important consideration in most intercept problems is the direction or orientation of the LOS between one vehicle and another. A simple example of the calculation is a description of relative target position by a pilot in a cockpit in terms of numbers on a clock face (12 o'clock, straight ahead; 3 o'clock, directly to the right; etc.) with elevation designated as high or low. Obviously, this

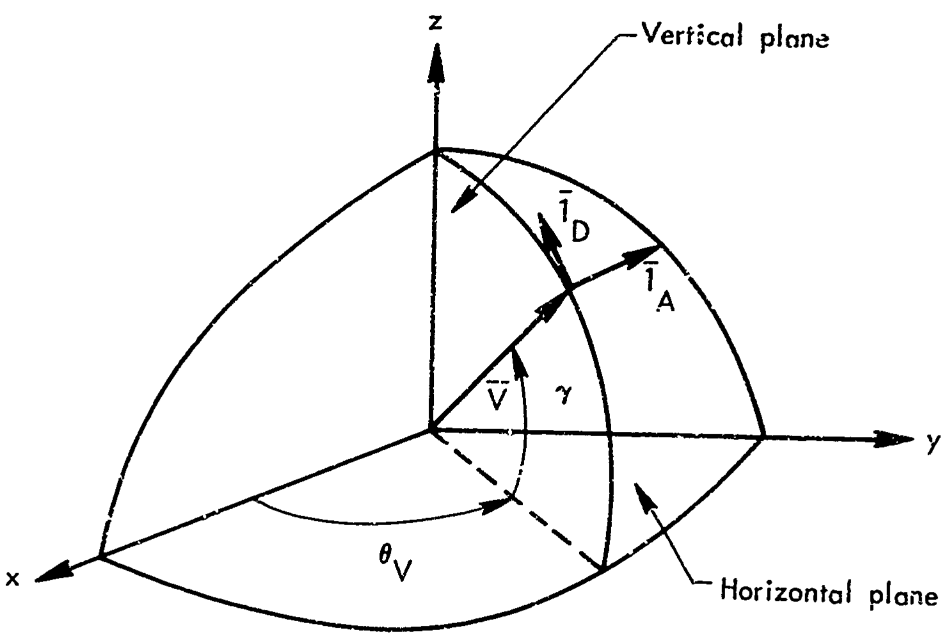


Fig. 5—Unit vectors normal to velocity vector

description would require alteration if the aircraft were suddenly to make a substantial change in attitude or orientation by pitching, rolling, or yawing. Similarly, TACTICS calculates and prints out azimuth (clock-face position) and elevation((high-low) orientation of the LOS from each vehicle relative to the other two. It also prints out a bearing angle defined here as the angle between the LOS and the longitudinal axis of the vehicle. Azimuth angle is measured in the plane formed by the right wing and the longitudinal axis as a pilot would view the relative geometry. Similarly, elevation angle is measured in the plane formed by the longitudinal axis and the line through the top of the cockpit. Azimuth angle ranges from 0 to ±180 deg, with positive to the right. Elevation angle ranges from 0 to ±90 deg, with positive upward, as shown in Fig. 6. Note that the bearing angle is defined as the total angle between the LOS and longitudinal axis and hence is independent of roll angle (see Appendix A for definition); this is a very useful feature, since it is confusing to visualize azimuth and elevation angles when a vehicle is maneuvering.

For those problems which involve sensors—e.g., infrared, radar, or optical—it is likely that constraints will be imposed on maximum values of elevation and on azimuth look angles due to mechanical, electrical, or optical limits. Provisions have accordingly been made to input any such constraints as initial-condition data.

Mathematical details and derivations pertaining to aircraft attitude or orientation angles and elevation, azimuth, and bearing-angle geometry are contained in Appendix A.

In summary:

- Associated with each vehicle are two LOS oriented toward the other two vehicles.
- The vehicle's coordinate system may be regarded as a longitudinal axis through the airframe, a line through the right wing, and a line through the top of the cockpit.
- The orientation of the LOS with respect to this coordinate system is resolved in terms of azimuth, elevation, and bearing angles.

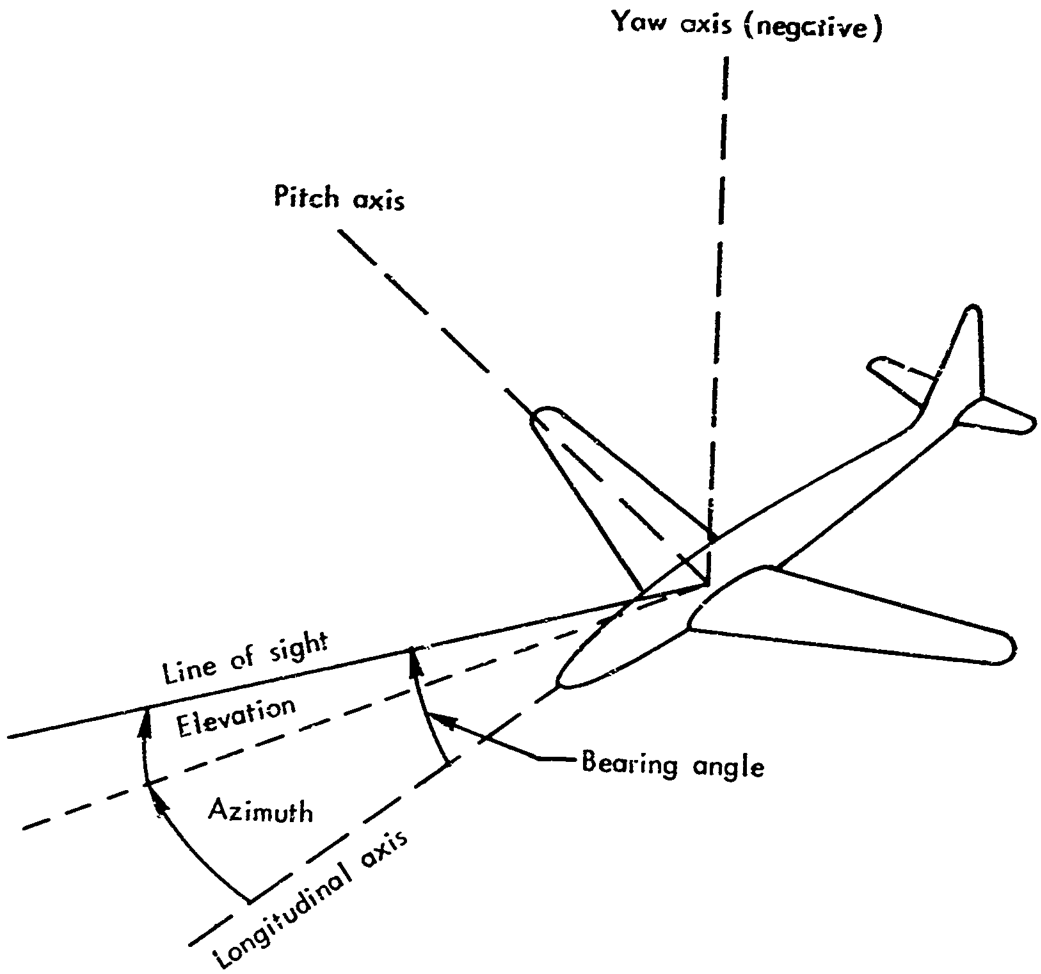
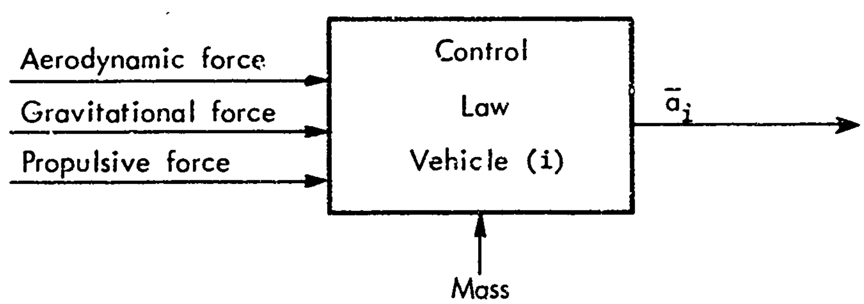


Fig. 6 — Elevation, azimuth, and bearing-angle geometry

IV. DEFINING THE FORCES AND ACCELERATIONS

CONTROL LAWS

TACTICS integrates the equations of motion defined by the three components of net acceleration associated with each of the vehicles. A resultant vector force $\overline{F}$ will define an acceleration $\overline{a}$. The forces of primary interest which will sum to this resultant force $\overline{F}$ may be categorized as (1) gravitational, (2) aerodynamic lift and drag, and (3) propulsive. If we assume that these forces are defined in magnitude and direction, it is a straightforward procedure to add them, resolve them into components, and determine the net acceleration. Each control-law subroutine may be considered as a modular unit where this process or its equivalent is performed, the output being the three components of net acceleration applicable to a particular vehicle, as shown schematically in the block diagram below.



The important point to note in this diagram is the correspondence of the net sum of forces to the net acceleration. Indeed, there are many control laws in which the net acceleration is by definition the starting point, and all control-law computations are primarily concerned with finding the correspondence between the forces that would be necessary to create such an acceleration.

To illustrate, consider a hypothetical control law in which all components of net acceleration are by definition zero. Assume that propulsive and gravitational forces are determined. The function of

the control law in this simple case is to determine the aerodynamic lift and drag forces necessary to guarantee the postulated output condition, i.e., zero acceleration.

LATERAL ACCELERATION AND ITS COMPONENTS

Many reference texts differ on the definition and usage of the term "lateral acceleration." Throughout this Memorandum, it is defined as a vector quantity in a plane normal to the velocity vector or flight path of the vehicle. Since the velocity vector $\overline{V}$ may be oriented in any direction in the general case, the above definition does not constrain the lateral acceleration vector to any particular direction or to any plane other than the one normal to $\overline{V}$.

Referring to Fig. 5, $\overline{1}_A$ and $\overline{1}_D$ are also by definition in the same plane normal to $\overline{V}$. Accordingly, it is convenient to resolve the lateral acceleration $\overline{a}$ into two components, a_h and a_V.

$$a_h = \overline{a} \cdot \overline{1}_A \qquad (7)$$
$$a_V = \overline{a} \cdot \overline{1}_D$$

By definition, the component a_h now represents a turning acceleration in the horizontal plane and a_V represents a climbing or diving acceleration in the vertical plane (caution: a_V is not necessarily in the vertical direction). At this point, it is necessary to distinguish between the specified or commanded values of $\overline{a}$ and its components (a_h and a_V) and the modified or output values. In simulating guidance of aircraft or missiles, real-world considerations often necessitate the modification of commanded values, because of constraints such as structural or aerodynamic limitations, time lag, and noise. This subject will be discussed further in Section V.

For the moment, let us assume that commanded values are constrained or modified so that output values result. With subscripts used to denote the difference, a commanded value of lateral acceleration is designated as $\overline{a}_C$ with components a_{Ch} and a_{CV}, and a modified or output

value as $\overline{a}_o$ with components a_{oh} and a_{oV}. The third component, $\dot{V}$, is assumed to be a function only of propulsion and aerodynamic drag forces with no constraints or modification. The total resultant acceleration $\dot{\overline{V}}$, describing the motion or trajectory of a vehicle represented as a point mass, is then

$$\dot{\overline{V}} = a_{oh}\,\overline{1}_A + a_{oV}\,\overline{1}_D + \dot{V}\,\overline{1}_V \tag{8}$$

where $\overline{1}_V$ is the unit vector along $\overline{V}$ and $\dot{V}$ is the time rate of change in the magnitude of $\overline{V}$. Expressed in inertial Cartesian ($\ddot{x}$, $\ddot{y}$, $\ddot{z}$) form, which is more convenient for numerical integration of the equations of motion, the components are

$$\begin{aligned}
\ddot{x} &= \dot{\overline{V}} \cdot \overline{i} = a_{oh}\,1_{Ax} + a_{oV}\,1_{Dx} + \dot{V}\,1_{Vx} \\
\ddot{y} &= \dot{\overline{V}} \cdot \overline{j} = a_{oh}\,1_{Ay} + a_{oV}\,1_{Dy} + \dot{V}\,1_{Vy} \\
\ddot{z} &= \dot{\overline{V}} \cdot \overline{k} = a_{oV}\,\overline{1}_{Dz} + \dot{V}\,1_{Vz}
\end{aligned} \tag{9}$$

where $\overline{i}$, $\overline{j}$, and $\overline{k}$ are the unit vectors along the x, y, z-coordinate frame axes, respectively, and 1_{Ax}, 1_{Dx}, etc. are the components of the unit vectors given by Eq. (6).

To represent the flight of three vehicles in motion simultaneously, 18 differential equations must be integrated, 9 involving the accelerations $\dot{\overline{V}}(i)$ and 9 the velocities $\overline{V}(i)$. TACTICS expresses these differential equations in two different forms: One uses the flat-earth representation (less complex and faster) and the other uses a round rotating or nonrotating earth (essential for space applications, long ranges, or high speeds). The expressions in Eq. (9) are the basic acceleration equations which are integrated for the flat-earth representation. Further details on numerical integration methods, round-earth form of the equations, coordinate transformations, and derivations are included in Appendix B. Next, the correspondence between net lateral acceleration and the gravitational, propulsive, and aerodynamic forces which cause this acceleration will be considered. At

this point it is important to stress that a vehicle maneuver may be defined in either of two ways, depending on the particular problem. A complete force-acceleration correspondence must be established by specifying either (1) the lateral acceleration and sufficient other information about the forces, or (2) the forces and sufficient other information about the lateral acceleration.

FORCES AND FORCE RESOLUTION

The various forces assumed to be acting on the center of gravity (c.g.) of each vehicle are shown in Fig. 7. In vector form, the summation of these forces is

$$m\dot{\overline{V}} = m\overline{g} + \overline{L} + \overline{T} + \overline{D} \tag{10}$$

where $\dot{\overline{V}}$ = vehicle acceleration

m = mass of the vehicle ($m = W/g$)

$\overline{g}$ = acceleration due to gravity (unit vector in the $-z$ direction)

$\overline{L}$ = lift force (defined as normal to the velocity $\overline{V}$)

$\overline{T}$ = resultant thrust force

$\overline{D}$ = drag force (unit vector in the $-\overline{V}$ direction)

This is a general vector expression where the positive or negative signs are accounted for by the orientation of the unit vectors associated with each term. For notational convenience, the i subscript which refers to a particular vehicle has been omitted. The above equation may be resolved and separated into two equations, one representing the lateral acceleration normal to $\overline{V}$ and the other representing the $\dot{V}$ acceleration in the direction of $\overline{V}$. Before doing this, certain important definitions or assumptions are necessary:

- The thrust force $\overline{T}$ acting through the c.g. is also coincident with the longitudinal body axis of the vehicle.
- The angle of attack α is taken to be the angle between $\overline{T}$ and the tangent to the flight path $\overline{V}$.
- All maneuvers consist of "coordinated turns," defined by the condition that a plane passed through the longitudinal body axis and including the yaw axis must also include the vector $\overline{V}$.

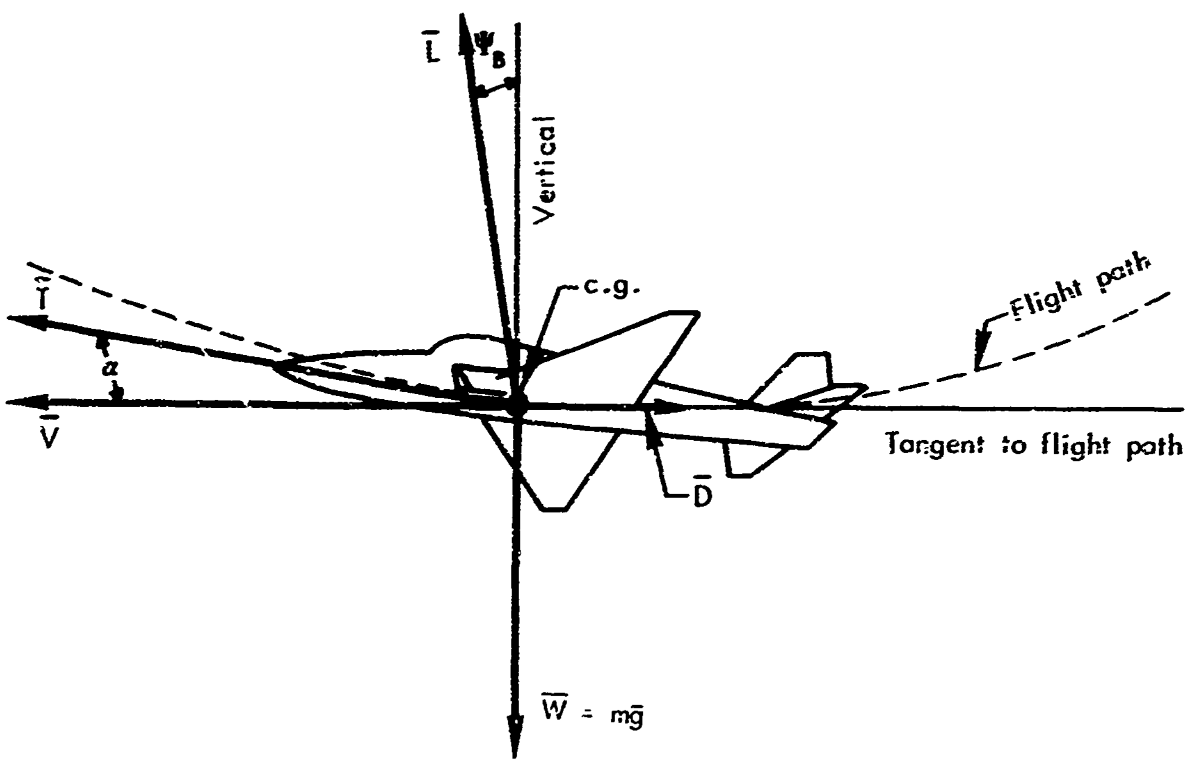


Fig. 7 — Coordinated turn in level flight

The situation for a coordinated turn in level flight is shown in Fig. 7. The vectors $\overline{L}$, $\overline{T}$, and $\overline{V}$ are co-planar and $\overline{T}$ is coincident with the longitudinal axis.

For the particular case of a level-flight turn, the bank angle ϕ_B is shown to be the angle between $\overline{L}$ and the vertical, but in the general case it is the angle between $\overline{L}$ and the reference vector $\overline{1}_D$ shown in Fig. 5. (Note that this is a wind axis or velocity reference, since $\overline{1}_D$ is normal to $\overline{V}$.) With these definitions and assumptions in mind, Eq. (10) may now be resolved into two scalar equations by summing forces and accelerations along $\overline{V}$ (direction $\overline{1}_V$) and in a plane normal to $\overline{V}$, using the symbol $\overline{1}_1$ to denote the direction of this resulting normal or lateral acceleration (see page 22 for definition of lateral acceleration).

$$\dot{V} = \frac{g}{W} (T \cos \alpha - D - W \sin \gamma) \tag{11}$$

$$a_o = \left[\frac{g}{W} (\overline{L} + \overline{T} \sin \alpha) + \overline{g}\right] \cdot \overline{1}_1 \tag{12}$$

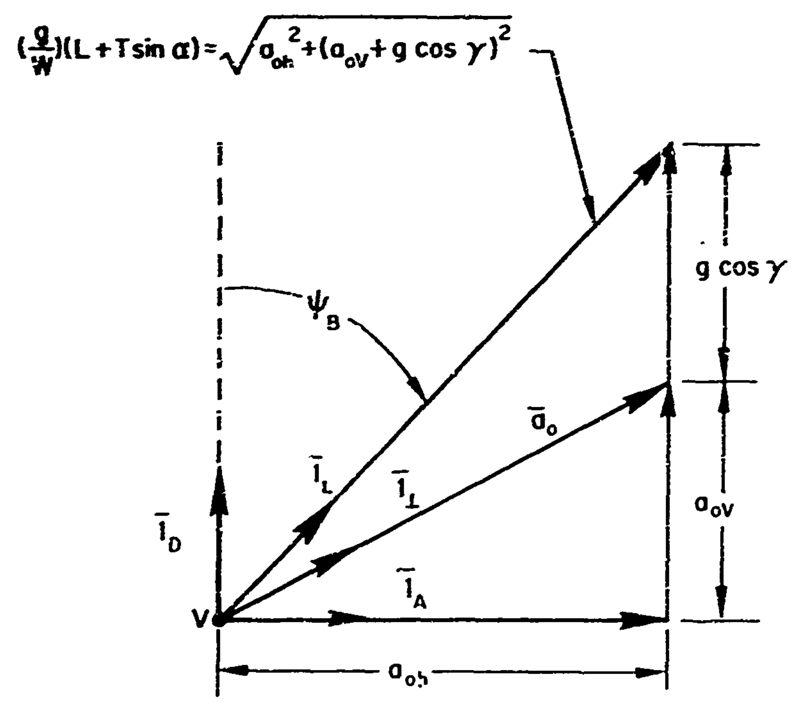
where a_o is the absolute magnitude of net lateral acceleration $\overline{a}_o$ and α is the angle of attack, as defined above.

Two examples will illustrate the basic relationships in order to avoid possible misunderstanding of definitions or terminology.

Case 1: Determination of Forces Required For a Specified Lateral Acceleration

Assume a simulation of a climbing turn arbitrarily defined by a lateral acceleration component a_{oh} in the horizontal plane and a positive a_{oV} component in the vertical plane. If the maneuver is initiated from level flight (i.e., $\overline{V}$ in the horizontal plane), the a_{oV} component and $\overline{1}_D$ will initially be oriented toward the vertical direction but not thereafter. Figure 8 shows the vectors and their scalar magnitudes. The total net lateral acceleration will be

$$a_o = \sqrt{a_{oh}^2 + a_{oV}^2} \tag{13}$$



Plane normal to V (V out of page)

Fig 8—Bank angle and lateral acceleration components

The normal force F_n, defined as the sum of aerodynamic lift and propulsive forces normal to $\overline{V}$, is

$$F_n = L + T \sin \alpha = \frac{W}{g}\sqrt{(g \cos \gamma + a_{oV})^2 + a_{oh}^2} \tag{14}$$

The direction of the vector $\overline{L}$ is given by

$$\overline{1}_L = \frac{W\left[a_{oh} \ \overline{1}_A + (a_{oV} + g \cos \gamma) \ \overline{1}_D\right]}{g \ F_n} \tag{15}$$

Note that the gravitational force component normal to $\overline{V}$ is associated with the g cos γ term.

Assuming α small, the initial ($\gamma = 0$) lift force L required for the maneuver is

$$L = \frac{W}{g}\sqrt{a_{oh}^2 + (a_{oV} + g)^2} \tag{16}$$

The direction of $\overline{a}_o$ is

$$\overline{1}_1 = \frac{(a_{oh} \ \overline{1}_A + a_{oV} \ \overline{1}_D)}{a_o} \tag{17}$$

and the bank angle is

$$\psi_B = -\sin^{-1}\left(\frac{a_{oh}}{L + T \sin \alpha} \cdot \frac{W}{g}\right) \tag{18}$$

The minus sign is arbitrarily assigned to make ψ_B negative for a left turn, i.e., a_{oh} is positive for θ_V increasing (shown in Fig. 4). Accordingly, in terms of bank angle the components of $\overline{a}_o$ become

$$a_{oh} = -\frac{g}{W}(L + T \sin \alpha) \sin \psi_B \tag{19}$$

$$a_{oV} = \frac{g}{W}(L + T \sin \alpha) \cos \psi_B - g \cos \gamma \tag{20}$$

The force F_n is defined as the force acting in the direction of the lift vector $\overline{L}$. Expressed in g's, it is

$$\frac{F_n}{W} = \frac{L + T \sin \alpha}{W} \tag{21}$$

From Fig. 5,

$$\cos \psi_B = \frac{a_{oV} + g \cos \gamma}{F_n} \left(\frac{W}{g}\right)$$

Hence,

$$\frac{F_n}{W} = \frac{a_{oV} + g \cos \gamma}{g \cos \psi_B} \tag{22}$$

For a horizontal turn in level flight, as shown in Fig. 7, this reduces to the simple expression

$$\frac{F_n}{W} = \frac{1}{\cos \psi_B} \tag{23}$$

Case 2: Determination of Acceleration Components Resulting From Specified Forces

Assume that aerodynamic and propulsive forces may be determined utilizing some prescribed control law and that it is necessary to determine the acceleration vector components a_{oh}, a_{oV}, and V from Eq. (10), which is repeated below:

$$\dot{\overline{V}} = \overline{g} + \frac{g}{W} (\overline{L} + \overline{T} + \overline{D}) \tag{24}$$

Ignoring for the moment the effects of time lag, aerodynamics, or structural constraints and referring to Eqs. (6), the acceleration components are

$$\begin{aligned} \dot{V} &= \dot{\overline{V}} \cdot \overline{1}_V \\ a_{oh} &= \dot{\overline{V}} \cdot \overline{1}_A \\ a_{oV} &= \dot{\overline{V}} \cdot \overline{1}_D \end{aligned} \tag{25}$$

The orientation of the lateral acceleration vector $\overline{a}_o$ is given by Eq. (16).

AERODYNAMIC FORCES

Aerodynamic computations may involve a large number of input parameters corresponding to the flight characteristics of a particular airframe. On the other hand, there are trajectory simulation problems where aerodynamic effects are not pertinent and no aerodynamic computations are necessary for a particular vehicle. An example is a point-mass target moving along a path where the motion is arbitrarily postulated, say, a straight-line path at constant speed.

In the general case, expressions for force magnitudes (lift L and drag D) are

$$L = C_L A q$$

$$D = C_D A q \qquad (26)$$

$$q = \tfrac{1}{2}\rho V^2$$

where C_L = lift coefficient
C_D = drag coefficient
A = aerodynamic reference area of the vehicle (ft^2)
q = dynamic pressure (lb/ft^2)
ρ = air density ($slug/ft^3$) (see Appendix I for a definition of model atmosphere)

In simulating the flights of specific aircraft designs, provisions have been made for incorporating tables to describe the interactions among lift coefficient C_L, drag coefficient C_D, angle of attack α, maximum lift coefficient $C_{L_{max}}$, and Mach number. Interpolation routines are employed to provide the equivalents of functional relationships. Alternatively, provisions have been made for using the following analytic expressions:

$$C_L = \frac{dC_L}{d\alpha}(\alpha - \alpha_o)$$

$$C_D = C_{D_o} + \frac{dC_D}{d(C_L^2)} C_L^2 \qquad (27)$$

where $dC_L/d\alpha$ = slope of the $C_L = F(\alpha)$ curve

α_o = zero-lift angle of attack (deg)

C_{D_o} = zero-lift drag coefficient

$dC_D/d(C_L^2)$ = a coefficient used with a parabolic function for drag coefficient, i.e., $C_D = F(C_L^2)$ (may be treated as a constant or a function of Mach number)

Details on how to provide and use tabulated values and on how to select the various options available are provided in Sections IX through XIII and Appendix D of this Memorandum.

PROPULSIVE FORCES

As previously mentioned, propulsive forces are assumed to act on the c.g. of the vehicle and to be coincident with the vehicle's longitudinal axis. As in aerodynamic computations, tables may be incorporated to describe the interrelationships between military and afterburner thrust, fuel flow, and Mach number. There are also provisions for reading in constant values for thrust as input data. A varying thrust condition may be simulated by the simple expedient of multiplying each thrust value by a "throttle" parameter (normally set to 1.0 unless otherwise specified in PØLICY). Additional flexibility is also available by incorporating thrust values or functional relationships in maneuver subroutines specialized to the particular problem or to vehicle characteristics.

GRAVITATIONAL FORCES

TACTICS is automatically set for the simplest assumption of a gravitational force—a flat earth causing 1 g or 32.174 ft/sec^2 acting downward in the negative z-direction. However, options are provided for considering the more complicated cases of a round earth, rotating or nonrotating, so that centrifugal, Coriolis, and inverse square law effects may be included if pertinent to the problem, e.g., in space applications and hypersonic flight. The basic expressions for the flat-earth representation are given by Eq. (9), and the gravitational

force term appears in a rather straightforward way (which is to be expected, since $\overline{g}$ is treated as a constant vector, i.e., in magnitude and direction) in Eqs. (11) through (16). All expressions and derivations relating to the round-earth representation are discussed in Appendix B because of the large number of details involving coordinate transformations and reference vectors. The gravitational acceleration in this case is taken to be rotating and varying in magnitude, as

$$\overline{g} = \frac{\overline{F}_g}{m} = \frac{-\mu\overline{R}}{R^3} \tag{28}$$

where $\overline{F}_g$ = gravitational force

$\mu = 1.407645 \times 10^{16}\ ft^3/sec^2$

$\overline{R}$ = radius vector from the geocenter to the vehicle

$R = |\overline{R}|$

V. AERODYNAMIC, STRUCTURAL, AND TIME-LAG CONSTRAINTS

AERODYNAMIC AND STRUCTURAL CONSTRAINTS

As mentioned in Section IV, in simulating vehicle flight performance it is usually necessary to impose constraints or limits on the forces applied to the vehicle. Two primary limitations are (1) *aerodynamic*, i.e., a limitation on lift coefficient C_L to some specified value representing a boundary on flight stability, and (2) *structural*, i.e., a load limit imposed by possible damage to the airframe or components (or possibly to a human being). The aerodynamic constraint imposes the following condition on the magnitude of the normal force F_n (see Eq. (14)):

$$F_n \leqq C_{Lmax} Aq + T \sin \alpha_{max} \tag{29}$$

where C_{Lmax} is a maximum value for C_L and α_{max} is a maximum value for α. Since C_L and α are functionally related, either one may be specified and the other calculated. Similarly, the structural constraint is

$$F_n \leqq \left(\frac{W}{g}\right) a_{Smax} \tag{30}$$

where a_{Smax} is the structural (or human) acceleration limit. It should be noted that by definition the force F_n is normal to the velocity vector $\overline{V}$. To be precise, the structural limit a_{Smax} should be considered as normal to the vehicle's longitudinal axis, which is separated from $\overline{V}$ by the angle α. In the TACTICS program and in the derivations which follow, this difference is ignored and a_{Smax} is assumed to be a *lateral* acceleration limit, by previous definition also normal to $\overline{V}$. Designating the specified or computed value of the normal force as F_{nC}, the aerodynamic limit as F_{na}, and the structural limit as F_{ns}, TACTICS takes the applied normal force F_n to be

$$F_n = \min(F_{nC}, F_{na}, F_{ns}) \tag{31}$$

where min is the minimum magnitude of the three values. When a constraint value (F_{na} or F_{ns}) is taken, the initial assumption is that the specified or computed direction of the commanded acceleration, i.e., the unit vector $\bar{1}_1$, remains constant but that the magnitudes of the forces and corresponding accelerations should be compatible with the constraining value. If a_{max} is designated as

$$a_{max} = \left(\frac{g}{W}\right) \min\ (F_{na}, F_{ns}) \tag{32}$$

the problem is to determine the corresponding magnitude of the lateral acceleration magnitude a_o. The applicable expression is

$$a_{max} = |a_o \bar{1}_1 + g \cos \gamma\ \bar{1}_D| \tag{33}$$

from which the following equation is derived:

$$a_o^2 + 2\ g \cos \gamma\ (\bar{1}_1 \cdot \bar{1}_D)\ a_o + (g \cos \gamma)^2 - a_{max}^2 = 0 \tag{34}$$

If the dot product is denoted as

$$\cos \delta = (\bar{1}_1 \cdot \bar{1}_D) \tag{35}$$

the solution for a_o is

$$a_o = -g \cos \gamma \cos \delta \pm \sqrt{a_{max}^2 - g^2 \cos^2 \gamma \sin^2 \delta} \tag{36}$$

To solve the quadratic equation, the following conditions must be satisfied:

- o The value of a_o must by definition be positive, since it represents the magnitude of a vector. (For multiple positive solutions the largest magnitude is taken.)
- o The discriminant should not be negative.

Difficulties in maintaining the direction of $\bar{a}_o$ (i.e., $\bar{1}_1$) arise when a_{max} becomes small (e.g., a wingless vehicle with small C_{Lmax}). These special conditions are handled as follows:

- If a positive solution for a_o is not possible, then the sign of $\cos \delta$ must be made negative, corresponding to a diving condition—i.e., the direction of a_{oV} must be reversed.
- If the discriminant becomes negative, then the value of $\sin \delta$ must be adjusted so that

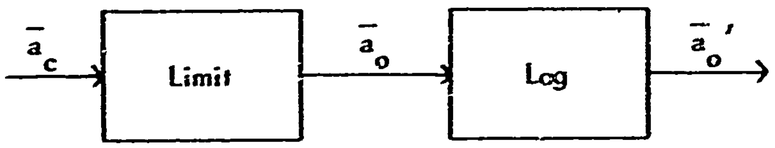
$$\sin \delta = \frac{a_{max}}{g \cos \gamma} \tag{37}$$

The net effect of either of the above alterations is to change the direction of the $\overline{I}_1$ vector. The resultant direction is given by

$$\overline{I}_1 = \sin \delta \, \overline{I}_A + \cos \delta \, \overline{I}_D \tag{38}$$

TIME LAG

In the diagram below, the block labeled "Lag" represents a transfer function to simulate time lag between an input command acceleration $\overline{a}_C$ and an output response $\overline{a}_o$.



The block labeled "Limit" represents the possible imposition of aerodynamic or structural constraints as discussed previously. Since certain arbitrary assumptions are involved in each of these processes, the purpose of the diagram is to emphasize the modular building-block form; alternate subroutines adapted to a particular problem may be substituted.

The time-lag transfer function describes the performance of the vehicle hardware mechanization in terms of control-system response, gyro prediction, tracking system, filter circuits, etc. Here the main generalization of the simulation occurs, for the question arises as to the pertinence of this aspect of vehicle performance to the problem solution. Unless a specific hardware design is being investigated,

the overall system response is usually represented by a product of simple first-order exponential functions of the form*

$$\frac{K_1}{1 + \tau s}$$

where s = the Laplace operator

τ = a time constant

K_1 = a constant gain factor

Until a more accurate and elaborate representation is needed, we have elected to use an input-output relationship of the form

$$a_o' = a_o \frac{K_1}{1 + \tau_1 s} \cdot \frac{K_2}{1 + \tau_2 s} \cdot \frac{K_3}{1 + \tau_3 s} \qquad (39)$$

This notation may be confusing, since a_o in the above equation may or may not be the commanded acceleration a_C depending on whether or not it is constrained. If all time constants are zero and no constraints exist,

$$a_C = a_o' = a_o$$

*Exponential solution of the form $K_1 e^{-t/\tau}$.

VI. GUIDANCE AND CONTROL LAWS

Generally, the call for a specified control law is conditional upon the fulfillment of one or more geometric, kinematic, or time conditions. The term "control law" as used here means defining a commanded lateral acceleration vector $\overline{a}_C$ associated with the vector velocity $\overline{V}$. For those laws involving aircraft or missile guidance, such as lead collision or proportional navigation, the calculation of the $\overline{a}_C$ vector quantity is dependent upon the kinematic state of the system (position, velocity, and perhaps acceleration); these may be classified as closed-loop guidance laws. On the other hand, there are open-loop control laws, e.g., turn, dive, climb, etc., in which the kinematic state of the system is not implicit within the law itself. This difference should become more apparent as these laws are summarized later.

Twenty-four different control laws are described in this Memorandum, many of which require lengthy explanations and derivations. For convenience, only a few illustrative examples are given below; a detailed listing of all control laws and their derivations is given in Appendix C.

A brief review of terms and of the basic applicable acceleration equations follows:

1. The commanded lateral acceleration vector $\overline{a}_C$ is by definition in a plane normal to $\overline{V}$ (see page 21) and is subject to possible aerodynamic, structural, and/or time-lag constraints resulting in the formulation of the vector $\overline{a}_o$ (see Section V).

2. The vector $\overline{a}_o$ is resolved into two components by the unit vectors $\overline{1}_A$ and $\overline{1}_D$ so that

$$\overline{a}_{oh} = \overline{a}_o \cdot \overline{1}_A$$

$$\overline{a}_{oV} = \overline{a}_o \cdot \overline{1}_D$$

3. If all forces—gravitational, propulsive, and aerodynamic—are specified, the *total* net acceleration $\dot{\bar{V}}$, consisting of the components a_{oh}, a_{oV}, and $\dot{V}$, may be determined. On the other hand, if a_{oV} and a_{oh} (gravitational and propulsive forces) are specified, the aerodynamic forces may be determined to calculate the corresponding component along $\bar{V}$ (see Eq. (25)).

4. The basic acceleration equations (flat-earth) are given by Eq. (9) in Cartesian ($\ddot{x}$, $\ddot{y}$, $\ddot{z}$,) coordinates, from which the velocities and positions are determined for each vehicle after numerical integration.

The guidance laws given in the illustrations below are almost all of the type in which the vector $\bar{a}_C$ and propulsive forces are specified and the corresponding aerodynamic forces are to be determined.

OPEN-LOOP CONTROL LAWS

Straight Flight

The commanded lateral acceleration $\bar{a}_C$ is zero. The vehicle will fly a straight-line path (but not necessarily "straight and *level*"). However, an acceleration or deceleration *along* this path may occur due to the thrust-drag relationship. The guidance law is

$$\begin{aligned} |\bar{a}_C| &= 0 \\ \bar{1}_1 &= \bar{1}_D \end{aligned} \qquad (40)$$

Captive Flight

This routine is used as a device to zero out computations and printed values for vehicles which are in captive flight. There are three modes:

- o Vehicle 2 locked to vehicle 1.
- o Any vehicle (1, 2, or 3) may be locked to the zero origin (i.e., zero position, velocity, and acceleration).
- o Vehicle 2 locked to vehicle 3.

Launch

This control law, which simulates the launch-boost phase of a missile flight, may be applied to any of the three vehicles. The call for "launch" is usually based on some criteria stated in PØLICY (e.g., range, range rate, geometry, accelerations, time, and—most importantly—combinations thereof). When this routine is called, the boost velocity ΔV must be specified as a constant. The commanded lateral acceleration is gravitational only:

$$\bar{a}_C = -g \cos \gamma \, \bar{1}_D \tag{41}$$

Left or Right Turn

The commanded lateral acceleration vector $\bar{a}_C$ is in the horizontal plane and has a constant value as specified when calling the routine(s). The two routines (left and right) are identical except for an algebraic sign corresponding to the direction of the turn ($\pm \bar{1}_A$). It was decided to specify the magnitude of the turning acceleration in terms of the resultant normal acceleration, which is expressed in g's or F_n/W (see Eq. (21)). Accordingly,

$$\bar{a}_C = g\sqrt{(F_n/W)^2 - \cos^2 \gamma} \qquad (\pm \bar{1}_A) \tag{42}$$

CLOSED-LOOP CONTROL LAWS

Proportional Navigation

The commanded lateral acceleration $\bar{a}_C$ is proportional to the space rate of rotation of the LOS between missile and target. Expressed in vector notation,

$$\bar{a}_C = \lambda V \bar{\omega}_r \times \bar{1}_V \tag{43}$$

where λ = the "navigation constant" (may be treated as either a constant or a variable)

V = vehicle speed

$\overline{\omega}_r$ = relative angular-rate vector as defined in Section III ($\overline{\omega}_r = \overline{\omega}_{ij}$ in Eq. (5))

$\overline{1}_V$ = unit vector along the missile velocity vector $\overline{V}$, i.e., $\overline{1}_V = \overline{V}/V$

The direction of the acceleration is defined by

$$\overline{1}_i = \overline{\omega}_r \times \frac{\overline{1}_V}{\omega_r}$$

The commanded acceleration $\overline{a}_C$ may be resolved into horizontal and vertical components by

$$\begin{aligned} \overline{a}_{Ch} &= a_C\left(\overline{1}_i \cdot \overline{1}_A\right) \\ \overline{a}_{CV} &= a_C\left(\overline{1}_i \cdot \overline{1}_D\right) \end{aligned} \tag{44}$$

Missile (X)

This routine is provided as an example of how to incorporate all significant aerodynamic, propulsive, and guidance characteristics of a hypothetical guided missile design into a single package. Numerical values which are unique to the configuration and which presumably will not be varied may be listed in the routine rather than supplied as input data for each simulation run (e.g., reference area A, initial weight W_o, burning rate $\dot{W}$, etc.). Moreover, specialized analytic functions, e.g., linear or polynomial curve fits, may be used for aerodynamic C_L, α, and C_D relationships as well as for propulsion characteristics. The guidance law for Missile (X) is a modified form of proportional navigation; again, for convenience, all necessary relationships are incorporated within the routine, which makes it unnecessary to call upon the proportional navigation routine. For further details see Appendix C.

VII. CONCLUSIONS

The basic framework, organization, input-output integration, flow, etc. of TACTICS are considered complete. However, in accordance with its purpose as a research tool, it is open-ended and subject to adaptation for each new problem; in this sense, it will never be complete. This adaptive process is simple and flexible because of the available options and because the main variables defining a problem can be treated externally by modular units (e.g., control-law and PØLICY subroutines). Hundreds of simulation problem runs, involving air-to-air combat, SAM, and ASM applications, have been performed with a wide variety of input-data, control-law, and PØLICY options. Experiments have also been performed for space applications using the two-body equation of motion. There is, however, no guarantee that the program will work perfectly for all cases in spite of all the check-out and operational experience. The number of possible configurations and combinations is extremely large, and it is unlikely that all will ever be tried. As with all computer problems, skepticism, intuitive reasoning, and cross-checking are necessary.

Part 2

OPERATING THE PROGRAM

VIII. INTRODUCTION

The sections which follow should be considered an operating manual for the program. The reader's familiarity with FORTRAN IV is assumed, but detailed knowledge or programming experience is not necessary. Section IX deals with formulating a POLICY subroutine composed of logical statements which dictate the control-law subroutines governing the flight of each vehicle, as explained in Section II. Section X explains how to set up a problem run by reading in initial-condition data and selecting options for integration methods, printout, table values, etc.

After the printed output sections are explained and illustrated in Section XI, several sample problems are described in detail in Section XII. The appendixes contain a list of FORTRAN instructions for calling optional subroutines (including an explanation of flags and argument variables), listings of subroutines, and an explanation of aerodynamic tables and formats. Careful study of the illustrative examples in Section XII is recommended, since they may serve as convenient guides for several types of problems.

IX. FORMULATING A PØLICY SUBROUTINE

In Section II the purpose and functional operation of the PØLICY subroutine were described. An illustration was given of an elementary flight plan calling for maneuvers such as climb, straight flight, and dive. Obviously, policy decisions based on some criteria are necessary to carry out such a flight p an, i.e., to dictate the transition from one maneuver to another. The FORTRAN notation for the elementary maneuver subroutines mentioned above is

- o CLIMB1 (I, GFØRC, IAERØ, ITHR)
- o STRFLT (I, IAERØ, ITHR)
- o DIVE1 (I, GFØRC, IAERØ, ITHR)
- o RTRN1 (I, GFØRC, IAERØ, ITHR)

where I = vehicle to which the law applies (1, 2, or 3)

IAERØ = an integer to indicate how aerodynamic computations are to be carried out

ITHR = an integer to indicate how propulsion computations are to be carried out

GFØRC = number of g's (F_n/W, as shown in Fig. 8 and given by Eq. (21)) required in the maneuver

The arguments IAERØ and ITHR specify whether table values, analytic expressions, or other alternatives are to be used (a complete description is given in Appendix D).

Assume that the flight plan mentioned above is required to simulate an actual takeoff of vehicle 1, an F-104 aircraft, and that the criteria that might be used are the following:

- o Time: TIME (straight flight at TIME = 0)
- o Speed: V(1,4) (sufficient speed for climb)
- o Flight-path angle: V(1,6) (to level off)
- o Altitude: R(1,3) (prior to turning)
- o Heading: V(1,5) (proper course angle)

The FORTRAN notation for position and velocity, expressed in both Cartesian and spherical coordinate systems, is as follows:

$$R(1,1) = x_1 \qquad R(1,4) = |\bar{r}_1|$$
$$R(1,2) = y_1 \qquad R(1,5) = \theta_{R1}$$
$$R(1,3) = z_1 \qquad R(1,6) = \varphi_{R1}$$

$$V(1,1) = \dot{x}_1 \qquad V(1,4) = |\bar{V}_1|$$
$$V(1,2) = \dot{y}_1 \qquad V(1,5) = \theta_{V1}$$
$$V(1,3) = \dot{z}_1 \qquad V(1,6) = \gamma_1$$

All FORTRAN notation for position, relative position, velocity, and relative velocity is parallel to the above.

A typical initial PØLICY statement for vehicle 1 at time zero might then be

```
CALL STRFLT (1,2,2)
```

where I = 1 (refers to vehicle 1)
IAERØ = 2 (refers to aerodynamic tables[†])
ITHR = 2 (refers to military thrust tables[†])

The successive statements might then be the following:

Climb at 0.5 g to 20-deg flight-path angle
- o IF (V(1,4) .GT. 120.0) CALL CLIMB1 (1,0.5,2,2)
- o IF (V(1,6) .GT. 20.*RAD) CALL STRFLT (1,2,2)[‡]

Begin leveling off at 1000 ft
- o IF (R(1,3) .GT. 1000.0) CALL DIVE1(1,0.5,2,2)

When within 0.1 deg of level flight, make a 0.5-g turn to a heading of 150.0 deg and resume straight flight
- o IF (V(1,6) .LT. 0.1*RAD) CALL RTRN1(1,0.5,2,2)
- o IF (V(1,5) .GE. 150.0*RAD) CALL STRFLT (1,2,2)

This elementary illustration defines a policy and flight path for one vehicle. For intercept trajectory problems, policies for two or

---

[†]F-104 tables will be loaded for vehicle 1.

[‡]The symbol RAD is used to convert degrees to radians.

more vehicles must be similarly defined. Omitting a reference to a vehicle (I = 1, 2, or 3) in PØLICY results in a zero acceleration definition for that vehicle, which may be appropriate for constant- or zero-velocity (e.g., ground-target) cases. Section XII presents more typical and complex examples of PØLICY subroutines with sample problems.

So far this subject has been discussed in the context of choices to be made within the PØLICY routine. Numerous other options may be selected by reading in flags or constants as part of initial-condition data. However, it is sometimes desirable to override these initial instructions in PØLICY if during a problem run a situation arises that requires, perhaps, a change in frequency of printout or integration step size. The following subsections list possible options in addition to those mentioned above.

STOPPING THE PROGRAM

The program automatically terminates after finding the closest miss distance between vehicles 2 and 1 or 2 and 3 unless otherwise specified by the flag IMISS. If no missile is launched, the program stops when running time (TIME) becomes greater than maximum specified time (TØTAL); this value is set in initial-condition data or in PØLICY.

If the program is to continue after finding the miss distance, flag IMISS = 1 must be set in PØLICY. This is used when a second missile is to be launched or if the vehicles are to continue on their trajectories (IMISS must be reset for every launch). For example, if the user wants the program to stop after finding the second missile's miss distance, IMISS must be reset to zero.

In summary:

- IMISS = 0. The program finds the closest miss distance of the missile and then stops.
- IMISS = 1. The program continues after finding the miss distance. IMISS must be reset for each launching.
- If no missile is launched, the program stops when time is greater than TØTAL (DATA 64).

INTEGRATION

Four different types of integration can be used (see Section XIII). The selection is made by setting the JINTEG flag as an initial condition (DATA 122).

If JINTEG = 0, variable-step Adams-Moulton predictor-corrector integration is used. Other values which are necessary when using this type of integration are ERTEST and HMIN. ERTEST (DATA 123) is the truncation error test for variable integration; if not specified by the user, it is automatically set at 1.0E-05. HMIN (DATA 135) is the minimum step size for the integration. The program sets HMIN = DTØ (the initial integration step set in DATA 136) unless otherwise specified.

If JINTEG = 1, fixed-step Runge-Kutta integration is used; if JINTEG = 2, fixed-step Adams-Moulton integration is used. In these cases the step size depends on the value read in for DTØ. If JINTEG = 3, the program integrates on a variable step size, controlled to allow printout exactly at specified intervals.

In summary:

- o JINTEG = 0: Variable-step Adams-Moulton integration.
- o JINTEG = 1: Fixed-step Runge-Kutta integration.
- o JINTEG = 2: Fixed-step Adams-Moulton integration.
- o JINTEG = 3: Variable-step Adams-Moulton integration with exact printout.

NUMBER OF VEHICLES USED

If the problem does not use one or more of the vehicles, its printout values can be set to zero by calling CAPFLT(I,MODE), where I indicates which vehicle is not to be used and MODE is set at 2. (See Example 3 in Section XII.)

TIME LAGS

Subroutine LAG represents a transfer function simulating the time lag between input command ACØM and output response AØUT. This function

may vary considerably in complexity, from a simple one-to-one correspondence to a highly complicated presentation of a missile guidance and control loop. The degree of realism (and hence complexity) depends, of course, on the specific problem and its significance to the final results. Accordingly, the LAG subroutine should be considered a flexible module which can be modified for a particular problem. A currently available LAG routine represents the time response of a vehicle as the product of as many as three first-order (exponential) time lags. Unless specified by the user, no time lag is used; AØUT = ACØM unless structurally (ASMAX) or aerodynamically (CLMAX) constrained. In order to introduce time delay, flag TAU(I) must be set in initial-conditions data to indicate the number of time lags desired, and TAU(I,J) must be set equal to the value of the time lags. See DATA 65-78 in Appendix E for details on reading the time constants. Example 3 in Section XII is an illustration of the way a time lag is used for a SAM.

RECALL

This feature enables the recall of the missile once it has been launched. For details on the use of the recall option, see Fig. 24 and Example 4 in Section XII.

RESTORE

The restore option enables the user to restore all numerical values existing at launch time if a hit or miss has occurred. See Fig. 27 and Example 5 in Section XII for details on its use.

ROUND EARTH, ROTATING OR NONROTATING

The FORTRAN flags IRT, IRF, IRØT8, IPRINT(I), INERF, and INERT are used to select the round-earth options. In defining satellite motion, velocity components are usually given with respect to a nonrotating, inertial coordinate frame. However, in defining aircraft or missile motion it is convenient to use velocity components expressed

with respect to a local earth-fixed rotating frame of reference. Of course, if the earth is considered as nonrotating, there is no difference.

These options are controlled as follows:

- o IRØT8 = 0: Nonrotating earth.
  IRØT8 = 1: Rotating earth.
- o INERF = 0: Vehicle 1 (fighter) velocity expressed in relation to local, earth-fixed, rotating frame.
  INERF = 1: Vehicle 1 velocity expressed in relation to local, earth-fixed nonrotating inertial frame.
- o INERT = 0 or 1: Corresponding statements apply to vehicle 3 (target) velocity.

The location of the origin of the local earth-fixed frame is specified by initial-condition data in terms of latitude LATO and longitude LØNGO. This is essential to properly account for rotational effects. The x-, y-, and z-axes of this coordinate frame are taken to form a right-handed system oriented in the following way: The z-axis is coincident with the local vertical or radius of the earth at the point LATO, LØNGO; the y-axis is in the local horizontal plane directed eastward; and the x-axis is directed southward or northward. Initial-condition position data for each vehicle must be supplied in terms of this local frame of reference, and there are three coordinate systems which might be used:

- o Cartesian (x, y, z) coordinates with respect to the origin defined by LATO, LØNGO (set IRF or IRT = 0 and use DATA 2-4 or 41-43 respectively).
- o Spherical (r, θ, φ) coordinates with respect to the origin defined by LATO, LØNGO (set IRF or IRT = 1 and use DATA 7-9 or 46-48 respectively).
- o Geocentric latitude, longitude, and altitude above a reference spheroid (set IRF or IRT = 2 and use DATA 111-113 or 116-118 respectively).

Mathematical details and derivations pertaining to the coordinate transformations and equations of motion are contained in Appendix B.

X. INPUT FORM AND ORDER OF INPUT

Input data include various problem constants, parameters, control flags (for options), and initial-condition values for position and velocity. There are two sections of data: (1) the aerodynamic tables for specific aircraft or missiles and (2) the main set, which gives initial conditions, flags, and constants. An input form is shown in Fig. 9 for convenience in specifying the main set of data.* A brief explanation is given here, but it is likely that the examples given in Section XII will more clearly illustrate the use of the form. Each line refers to a data card (the examples will show that in general most of the lines and spaces may be left blank).

The first line, i.e., the first data card, specifies whether aerodynamic tables are to be used for a vehicle. It gives values to the flags JVEH(I), I = 1,2,3. If JVEH(I) = 1, tables for vehicle (I) will be used; if JVEH(I) = 0, tables for vehicle (I) will not be used. (If not, the appropriate aerodynamic and propulsion constants may be entered in DATA 80-94.) The spacing for the JVEH card is as shown in Fig. 10. If tables are to be used, they are read immediately following the JVEH card, as shown in Fig. 11.

A title card describing the run (FØRMAT 12A6) is placed between the aerodynamic tables and the main set (this card is required). The locations (i.e., data numbers) and descriptions of all input data are given in Appendix E. However, the input form has spaces only for those locations most likely to be used, with spare spaces available at the bottom of the page. (There are provisions for 200 data locations, but only 143 are currently being used.)

AERODYNAMIC AND PROPULSION TABLES

The user is able to approximate the aerodynamics necessary to describe the flight of an aircraft by means of tables. If tables are used, they furnish such information as angle of attack and drag

---

*Figure 9 shows the input form reduced in size. A full-sized version is included at the end of the Memorandum for the reader's use.

TACTICS PROGRAM - INPUT FORM[a]

VEHICLE TABLE FLAGS (0 OR 1)[b]

5	12	18
#1	#2	#3

TITLE CARD[b]

5 10 15 20 25 30 35 40 45 50 55 60 65 70 72

MAIN SET OF INITIAL CONDITIONS

Columns: 1 4 6 15 18 20 29 32 34 43 46 48 57 60 62 71 74

Vehicle #1

Card					
0001	Cartesian (0)	x	y	z	Initial weight
0006	Spherical (1)	r	θ	φ	Reference area
0011	Cartesian (0)	ẋ	ẏ	ż	$a_{S_{max}}$ (lateral g's)
0016	Spherical (1 or 2)	V	θ_V	γ	Mach no. flag

Vehicle #2

Card					
0025	Initial weight	Reference area	$a_{S_{max}}$ (lateral g's)	Captive flight (0,1,2,3)	Maxim a range

Vehicle #3

Card					
0040	Cartesian (0)	x	y	z	Initial weight
0045	Spherical (1)	r	θ	φ	Reference area
0050	Cartesian (0)	ẋ	ẏ	ż	$a_{S_{max}}$ (lateral g's)
0055	Spherical (1 or 2)	V	θ_V	γ	

PRINT

Card					
0062	Initial time	Print interval	Termination time		

GUIDANCE (3 VEHICLES)

Card	Number of time constants	τ(1)	τ(2)	τ(3)	λ navigation constant
0065					
0070					
0075					

AERODYNAMICS AND PROPULSION (3 VEHICLES)

Card	$C_{L_{max}}$	C_{D0}	dC_D/dC_L^2	$dC_L/d\alpha$	α_0
0080					
0085					
0090					

Card	Military thrust	Afterburner thrust	Specific impulse	Burnout weight	Boost acceleration
0095					
0100					
0105					

ROUND-EARTH OPTION[c]

Card					
0110	Round-earth flag	Altitude (#1)	Longitude (#1)	Latitude (#1)	Velocity flag (#1)
0115	Rotation (1)	Altitude (#3)	Longitude (#3)	Latitude (#3)	Velocity flag (#3)
0120	Origin longitude	Origin latitude			

MISCELLANEOUS OPTIONS AND EXTRA INPUTS[c]

Card					
0122	Integration flag	No. of significant digits req.	Miss calculation R_{min}	θ_V aiming error (#1)	θ_V aiming error (#2)
0127	θ_V aiming error (#3)	γ aiming error (#1)	γ aiming error (#2)	γ aiming error (#3)	Ballistic coefficient (#1)
0132	Ballistic coefficient (#2)	Ballistic coefficient (#3)			
d -					

[a] The following units are used: distance (ft), time (sec), velocity (ft/sec or Mach no.), acceleration (g's), angles (deg), weight (lb), area (ft^2)

[b] This card required.

[c] Leave this section blank if option not used.

[d] Last data card must have a minus sign in column 1.

Fig. 9 — Input form

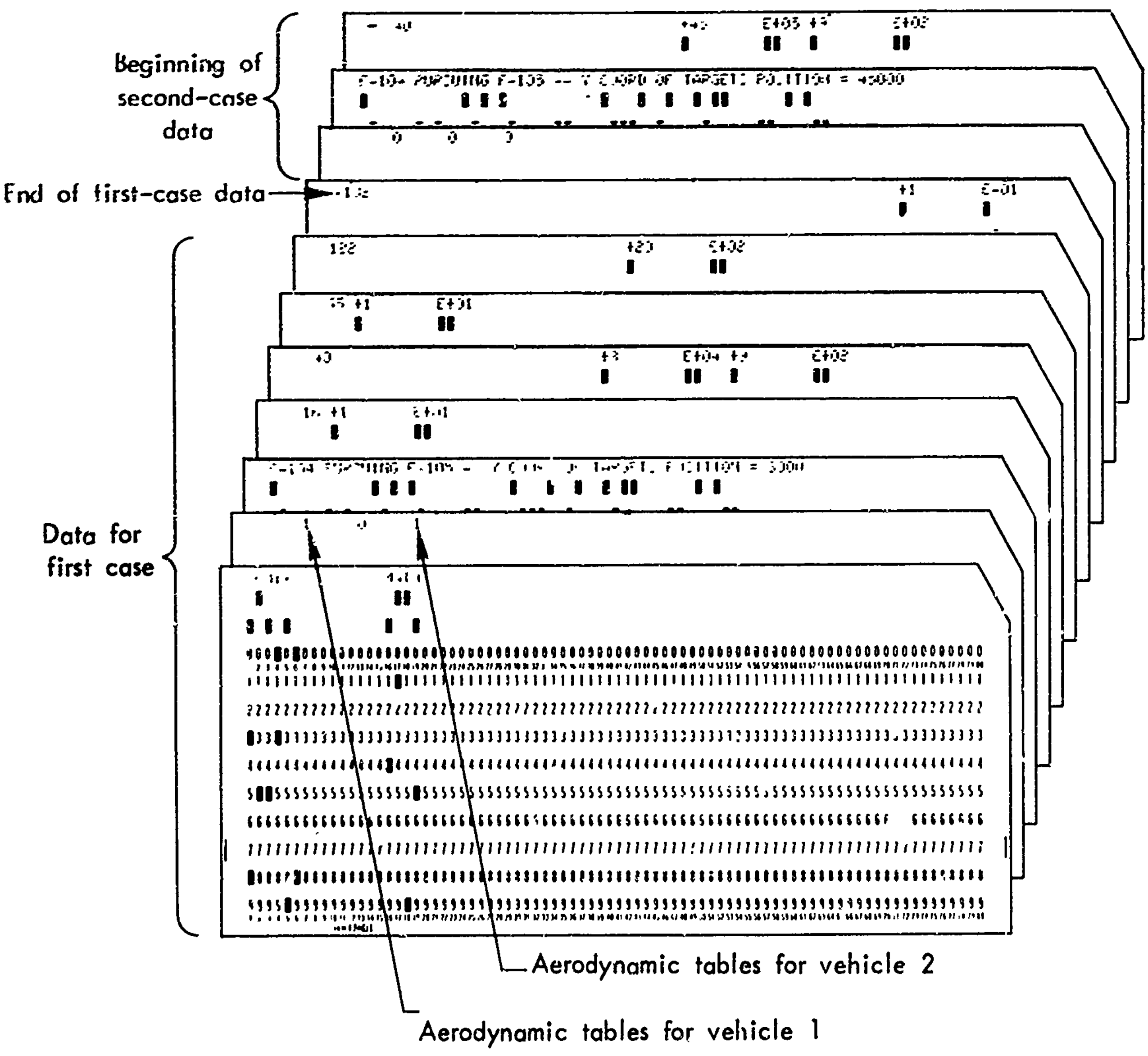


Fig. 10 — Cards specifying main data

Data for first run

JVEH flag card
Tables (if first card not all zeros)
Title card
Main set of data (DATA 1-143)

Data for second run

JVEH flag card
Tables (if different from first run)
Title card
Main data that differs from that of first run

Data for third run, etc.

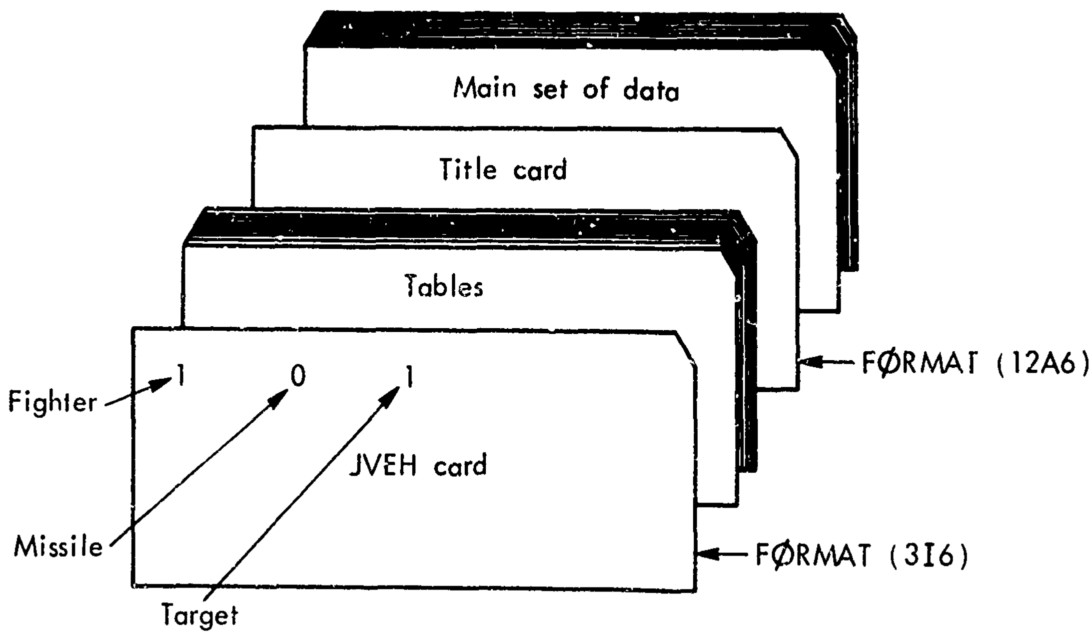


Fig. 11 — Order of input

coefficient as a function of lift coefficient and Mach number and thrust as a function of altitude and Mach number.

When calling a control-law subroutine, the flags IAERØ and ITHR are used to select the aerodynamic and propulsion operations pertaining to the choice of table values, analytic functions, or constants. Instructions for calling these options are given in Appendix D.

Appendix F describes the form and organization of aerodynamic and propulsion tables. A complete set of tables is also shown as an example in Fig. 35, Appendix F. The model atmosphere is calculated by analytic expressions (from Ref. 4) as given in Appendix I (there are no provisions for using table values for model atmosphere representation).

AERODYNAMICS AND PROPULSION WITHOUT TABLES

There are several possibilities for handling aerodynamic or propulsion calculations for the JVEH(I) = 0 option:

o Constants may be read in the appropriate locations (80-94) and Eq. (27) used for C_L, C_D, and α computations by specifying the argument IAERØ = 1 when calling the control-law subroutine. (See Appendix D for instructions.)

o Constants may also be used in appropriate locations (95-109) to specify values of propulsion and fuel flow characteristics, e.g., thrust, specific impulse, burnout weight, or boost acceleration. Thrust is set equal to the data value by using the ITHR flag as explained in Appendix D for ITHR = 3, 4, or 5. Vehicle weights are determined from the expression

$$W = W_o - \int \dot{W}\, dt$$

where $\dot{W}$ may be a value determined from fuel flow tables (ITHR = 1, 2) or a constant input value as specified in data locations 144-146 when using the ITHR = 3, 4, 5 option. However, in order to simulate a constant-acceleration boost phase of a missile, the program will automatically calculate vehicle weight from the expression $W = W_o \exp a_B(t-t_L)/gI$

where W = the current weight at time t

W_o = the initial weight

t_L = the launch time

a_B = the boost acceleration, considered to be an average value

I = the specific impulse of the rocket motor

This automatic alternative computation is initiated when an input data value for a_B is supplied in locations 99, 104, and 109 (as applicable to vehicle 1, 2, or 3). Provisions are also made for supplying missile burnout weights in locations 98, 103, and 108, but no computations are performed on these quantities within the main body of the program itself; the locations are provided merely as a convenience for formulating subroutines requiring this form of input data.

o For those cases where aerodynamic computations for a particular vehicle are not significant to the problem, it is not necessary to read in either tables or constants for that vehicle. The IAERØ = 3 argument is used (see Appendix D) and the vehicle is assumed to be a point mass moving at constant speed (but not necessarily at zero lateral acceleration).

o For specialized aerodynamic or propulsion characteristics (e.g., Missile (X), Appendix C) analytic expressions and necessary constants may be incorporated into the control-law subroutine.

MAIN SET OF DATA

The main set of data specifies such initial conditions as vehicle position and velocity, aerodynamic constants, structural and attitude limits, time lags, program flags (see Appendix H) and constants. Data numbers with corresponding program variables and descriptions are given in Appendix E. There are 143 separate entry spaces in the main data. If a value is not read into a space, that entry is automatically taken to be zero; therefore, only nonzero data need be specified.

The format for reading in the main data is (A1, I3, 5E14.8). The initial value on the card is the number of the first data entry on the card. Entries which follow this value on the card must be in sequence. A minus sign is placed in the first column of the last data card to indicate that the entries for that case are finished. Any data following

this card are for a new case (see Fig. 10). For sample data, refer to the numerical examples in Section XII.

Initial Position and Velocity Data

As shown on the input form, data spaces are available for those flags that indicate how the position and velocity of the vehicles are to be entered: in spherical or Cartesian coordinates or (in the case of position only) in latitude and longitude. Data entries which follow are for specifying the positions and velocities of vehicles 1 and 3. Vehicle 2 is initially assumed to be a missile attached to vehicle 1; however, data locations 30 through 35 (not shown on the input form) are available for setting initial position and velocity conditions separately for vehicle 2. That is, if data entries are made in any one of these locations (30-35), TACTICS will start simultaneously computing the trajectories of all three vehicles; otherwise, vehicle 2 is initially attached to 1 and computations will be performed for only two vehicles (unless or until there is a PØLICY call for launching 2).

Aerodynamic Constants

If analytic functions are to be used for aerodynamic computations, the equation constants, i.e., those applicable to Eqs. (26) and (27), must be entered. Data spaces are assigned for such constants, as shown on the input form.

Other Data

Spaces for structural and aerodynamic constraints are shown on the input form. Maximum elevation and azimuth angle gimbal limits may be specified by DATA 138-143. Up to three consecutive vehicle time lags can be entered (see Section V). Program flags and constants should be set; typically, these would specify minimum integration step, type of integration, and total time of run. If certain necessary values for program operation are not entered, TACTICS will automatically assume "default" values, print an informative message, and continue.

RUNNING MORE THAN ONE CASE

Any number of consecutive cases can be submitted for a single run. As indicated previously, the last card of the data for a case has a minus sign in the first column indicating the end of the case. To submit a second case, only data differing from values in the first case need be entered. For each case a JVEH card and comment card must be entered before the main data. With reference to Fig. 10, note that although the JVEH card contains all zeroes, the same tables are used in the second case as in the first.

XI. OUTPUT

The output from a typical computer run consists of labeled and unlabeled initial-condition data values and optional sections of printout. Initial-condition values will automatically be printed, but the printing of the optional sections must be specified in PØLICY. The printing of unlabeled data consists of a listing of all locations, DATA 1-200, as shown in Fig. 12. Labeled initial conditions for these values are as shown in Fig. 13. Table 1 lists sets of available optional output sections, the most important of which is the main set or "standard output" giving position, velocity, acceleration, weight information, and other basic quantities printed at specified time intervals; these intervals are controlled by the variable DTPØ (DATA 63). The other sections concern information about aerodynamics, attitude, etc. as indicated in Table 1. In order to specify these optional sections in PØLICY, the variable NPRINT is set equal to the required number of output sections.

A typical output specification in PØLICY would be the following:

NPRINT = 3: Three sections

IPRINT(1) = 1: Standard output

IPRINT(2) = 2: Aerodynamics

IPRINT(3) = 3: Attitude angles

A sample page of output for the NPRINT = 3 option is shown in Fig. 14. Most of the items shown are self-explanatory except perhaps the following:

1. The integration step size shown at the top of Fig. 14 is the last integration step taken before printout. If the JINTEG = 3 mode of integration is being used, two values will be printed: the step taken to reach printout time (TPØ) and the step the routine would have used if there had not been an immediate print requirement. (The step taken to reach printout must necessarily be the smaller of the two.)

2. The units used are feet for distance, degrees for angles, pounds for force and weight, g's for acceleration, and radians per second for angular rates.

*****INITIAL DATA VALUES*****

1	0.0	0.46190000E 04	-0.80000000E 04	0.20000000E 05	0.16990000E 05
6	0.0	0.0	0.0	0.0	0.19600000E 03
11	0.0	0.0	0.0	0.0	0.72999992E 01
16	0.10000000E 01	0.91999996E 00	0.90000000E 02	0.0	0.10000000E 01
21	0.0	0.0	0.0	0.0	0.0
26	0.0	0.0	0.10000000E 01	0.0	0.0
31	0.0	0.0	0.0	0.0	0.0
36	0.0	0.0	0.0	0.0	0.0
41	0.0	0.0	0.22000000E 05	0.33096000E 05	0.0
46	0.0	0.0	0.0	0.38500000E 03	0.0
51	0.0	0.0	0.0	0.72999992E 01	0.10000000E 01
56	0.86999995E 00	0.90000000E 02	0.0	0.0	0.0
61	0.0	0.0	0.50000000E 00	0.15000000E 02	0.0
66	0.0	0.0	0.0	0.40000000E 01	0.0
71	0.0	0.0	0.0	0.0	0.0
76	0.0	0.0	0.0	0.40000000E 01	0.0
81	0.0	0.0	0.0	0.0	0.0
86	0.0	0.0	0.0	0.0	0.0
91	0.0	0.0	0.0	0.0	0.0
96	0.0	0.0	0.0	0.0	0.0
101	0.0	0.0	0.0	0.0	0.0
106	0.0	0.0	0.0	0.0	0.0
111	0.0	0.0	0.0	0.0	0.0
116	0.0	0.0	0.0	0.0	0.0
121	0.0	0.30000000E 01	0.50000000E 01	0.10000000E 04	0.0
126	0.0	0.0	0.0	0.0	0.0
131	0.0	0.0	0.0	0.0	0.0
136	0.99999979E-02	0.0	0.0	0.0	0.0
141	0.0	0.0	0.0	0.0	0.0
146	0.0	0.0	0.0	0.0	0.0
151	0.0	0.0	0.0	0.0	0.0
156	0.0	0.0	0.0	0.0	0.0
161	0.0	0.0	0.0	0.0	0.0
166	0.0	0.0	0.0	0.0	0.0
171	0.0	0.0	0.0	0.0	0.0
176	0.0	0.0	0.0	0.0	0.0
181	0.0	0.0	0.0	0.0	0.0
186	0.0	0.0	0.0	0.0	0.0
191	0.0	0.0	0.0	0.0	0.0
196	0.0	0.0	0.0	0.0	0.0

Fig. 12 — Sample initial data values (unlabeled)

SAMPLE POLICY

**********INITIAL CONDITIONS**********

	RECT. COORD.			SPHER. COORD		
POSITION	X(FT)	Y(FT)	Z(FT)	R(FT)	THETA(DEG)	PHI(DEG)
FIGHTER	0.46190000E 04	-0.80000000E 04	0.20000000E 05	0.22030320E 05	-59.99890	65.20847
MISSILE	0.46190000E 04	-0.80000000E 04	0.20000000E 05	0.22030320E 05	-59.99890	65.20847
TARGET	0.0	0.0	0.22000000E 05	0.22000000E 05	89.99998	89.99998
RELATIVE RANGE						
FIGHTER-MISSILE	0.0	0.0	0.0	0.0	89.99998	0.0
FIGHTER-TARGET	-0.46190000E 04	0.80000000E 04	0.20000000E 04	0.94517266E 04	120.00113	12.21622
MISSILE-TARGET	-0.46190000E 04	0.80000000E 04	0.20000000E 04	0.94517266E 04	120.00113	12.21622

RELATIVE RANGE RATE OF CHANGE	(FT/SEC)
FIGHTER-MISSILE	0.0
FIGHTER-TARGET	-50.0888
MISSILE-TARGET	-50.0888

VELOCITY	X(FT/SEC)	Y(FT/SEC)	Z(FT/SEC)	V(FT/SEC)	THETA(DEG)	GAMMA(DEG)
FIGHTER	0.00	955.55	0.0	955.55	89.99998	0.0
MISSILE	0.00	955.55	0.0	955.55	89.99998	0.0
TARGET	0.00	696.37	0.0	896.37	89.99998	0.0
RELATIVE VELOCITY						
FIGHTER-MISSILE	0.0	0.0	0.0	0.0	89.99998	0.0
FIGHTER-TARGET	-0.00	-59.18	0.0	59.18	-90.00009	0.0
MISSILE-TARGET	-0.00	-59.18	0.0	59.18	-90.00009	0.0

FLAGS AND CONSTANTS

TIME	DTPU	TOTAL	JINTEG	DTO	ERTEST	HMIN	DTMIN
0.0	0.5000	15.0000	3.0000	0.0100	0.100000E-04	0.100000E-01	0.100000E-03
JATMGS	**KLAUN**	**KMTMAX**	**MINMR**	**TBURN1**	**TBURN2**		
1.0000	0.0	0.0	1000.0000	0.0	0.0		
ASMAX	**CLMAX**	**ALPHAO**	**ABOOST**	**AREA**	**WO**	**DVTH**	**DGAMMA**
7.3000	0.9332	0.0	0.0	196.0000	0.169900E 05	0.0	0.0
**********	**********	0.0	0.0	0.0	0.0	0.0	0.0
7.3000	0.9795	0.0	0.0	385.0000	0.330960E 05	0.0	0.0
CDOCON	**BCON**	**THCON**	**TABCON**	**TAU**	**LAMDAO**	**IMPLSE**	**SLOPE**
0.0	0.0	0.0	0.0	0.0	4.0000	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	4.0000	0.0	0.0
ELEVMX	**AZMAX**						
0.0	0.0						
0.0	0.0						
0.0	0.0						

Fig. 13 — Sample initial data values (labeled)

```
TIME =    0.5000 SEC        INTEGRATION STEP SIZE =  0.020000,  0.160000

                    REL.RANGE            REL.VEL.        ROOT(REL)                    VELOCITY     THETAV       GAMMA
FIGHTER-MISSILE   0.0                       0.0              0.0          FIGHTER      958.188      90.000       0.0
FIGHTER-TARGET    0.94265391E 04           59.8982         -50.644        MISSILE      958.188      90.000       0.0
MISSILE-TARGET    0.94265391E 04           59.8982         -50.644        TARGET       898.290      90.000       0.0

     OMEGAR            UMEGAB            ACOM        AOUT        AOUTA       AOUTD       GFORCE
0.0               0.0               0.0         0.0         0.0         0.0         1.00000
0.0               0.0               0.0         0.0         0.0         0.0         0.0
0.0               0.0               0.0         0.0         0.0         0.0         1.00000

     GAMDOT        THDOT*COS(GAM)        VDOT                                 EXTR               EXTR
0.0               0.0             0.52569275E 01                  0.0                 0.0                          STRFLT
0.0               0.0             0.52569275E 01                  0.0                 0.0                          CPFLTI
0.0               0.0             0.38320951E 01                  0.0                 0.0                          STRFLT

POSITION               X                  Y                  Z                          ABS. RANGE
 FIGHTER       0.46190000E 04    -0.75215586E 04    0.20000000E 05             0.21861129E 05
 MISSILE       0.46190000E 04    -0.75215586E 04    0.20000000E 05             0.21861129E 05
 TARGET        0.14749821E-03     0.44866675E 03    0.22000000E 05             0.22004570E 05

      MACH         MACHMX            THRUST            WEIGHT
    0.92254       1.68000        0.52363477E 04   0.16989059E 05
    0.92254       0.0            0.52363477E 04   0.0
    0.87186       1.76667        0.80474375E 04   0.33094754E 05

**********AERODYNAMICS**********

      CLMAX          CL          LIFT               CD          DRAG             ALPHA
     0.93970       0.14753     16800.11719        0.02158     2457.08545        2.06763
************       0.0             0.0            0.0            0.0            0.0
     0.98015       0.17812     32698.52344        0.02231     4095.91602        2.82196

*********ATTITUDE ANGLES**********

                     ELEV    AZIMUTH    BEARNG                ROLL     ROLL RATE     BANK          THRREL     PHRREL
FIGHTER-MISSILE      0.0      0.0        0.0      FIGHTER     0.0        0.0         0.0            0.0        0.0
FIGHTER-TARGET      10.5    -29.9       31.5      MISSILE     0.0        0.0         0.0          120.1       12.2
MISSILE-TARGET       0.0      0.0        0.0      TARGET      0.0        0.0         0.0          120.1       12.2
MISSILE-FIGHTER      0.0      0.0        0.0
TARGET-FIGHTER      -9.8    150.2      148.8
TARGET-MISSILE       0.0      0.0        0.0
```

Fig. 14 — Sample page of output for NPRINT = 3

3. GAMDØT and THDØT*CØS (GAM) are the angular rates of rotation of the vehicles' velocity vectors.

4. ØMEGAR refers to the angular rates of rotation of the LOS RREL(I,J). ØMEGAB output is provided for guidance laws that use a biased angular-rate term (see Appendix C).

5. An EXTR FØRMAT (E16.8) is provided for six extra quantities (middle right of standard output package). This is extremely useful for debugging and printing additional information.

6. A CLMAX or ASMAX print (adjacent to acceleration quantities) will occur whenever these respective limits are exceeded. If no input data value has been given to CLMAX or ASMAX, these values are assumed to be infinite (e.g., 10^6).

7. PHRREL and THRREL (bottom right) refer to the orientation angles φ and θ, respectively, of the LOS, i.e., RREL(I,J).

Table 1

OUTPUT SECTIONS

Section	Title	Description
1	Standard output	Position, velocity, acceleration, and other basic quantities
2	Aerodynamics	Lift, maximum lift and drag coefficients, angle of attack, lift, and drag
3	Attitude angles	Roll, bank, elevation, azimuth, and bearing angle
4	Round-earth coordinates	Latitude, longitude, and altitude
5	Initial conditions	Input data
6	Atmosphere	Air pressure, temperature, air density, speed of sound, and Mach number
9	Angular rates	Angular-rate vectors and unit vectors $\overline{1}_A$, $\overline{1}_D$, $\overline{1}_V$ and $\overline{1}_{\perp V}$

NOTE: Sections 7, 8, and 10 through 18 are blanks available to the user if different types of output are desired.

XII. EXAMPLES

Five examples indicating the type of program which can be run on TACTICS and the way the program can be used are given in this section. Each problem is defined by initial-condition input data and by a PØLICY subroutine which dictates the control laws governing the flight of each vehicle. The examples show how a PØLICY subroutine is set up and what input data are necessary for a particular problem. The examples selected cover the different options available in TACTICS by employing a number of the special features included in the program (e.g., restore and recall). These examples range from the simple case of vehicles flying maneuvers with no missile to the more complicated one of launching two missiles.

EXAMPLE 1: AIRCRAFT MANEUVERS, NO MISSILE

The flights of two vehicles in different maneuvers are simulated; no missile is involved. In setting up this example, as with all the others, two steps are necessary: (1) developing the PØLICY subroutine and (2) entering initial-condition data.

PØLICY Subroutine

The following POLICY statements define the problem:

Vehicle 1 (Constant-Speed Aircraft: No Aerodynamic or Propulsion Computations)

- o Straight flight.
- o After 0.1 sec, pull a 4-g climb.
- o When vehicle has climbed 30 deg, level off to horizontal and continue on straight flight.

Vehicle 2

- o Not used.

Vehicle 3 (Constant-Speed Aircraft: No Aerodynamic or Propulsion Computations)

- Fly straight flight.
- After 1 sec, perform a barrel roll pulling 4 g's and rolling 60 deg/sec.
- When roll is completed, perform 4-g diving turn with 130-deg roll.

These statements are translated into FORTRAN expressions to formulate the PØLICY subroutine. Figure 15 is a listing of the routine.

The type of output desired is first specified in PØLICY (see Section XI). In this example, only the standard output and attitude angles are to be printed, since no aerodynamic or propulsion computations are being performed.

Commands governing the first vehicle's flight are given. There are different control laws and criteria for changing maneuvers. For each maneuver the type of aerodynamic and thrust computation must be specified in the argument following the maneuver name (see Appendix D). For this example, no aerodynamic or propulsion computations are involved, so IAERØ = 3 and ITHR = 5.

CLIMB1 (simple climb) is called, specifying 4 g's in the argument, until vehicle 1 has climbed 30 deg. The criterion used in this case is V(1,6), the flight-path angle γ_1. STRLVL is called to level off the vehicle. Flag LEVEL(1) = 2 indicates that the aircraft is horizontal again.

Since the second vehicle is not being used, CAPFLT(2,2) is called, which sets all values pertaining to the vehicle equal to zero.

The motion of the third vehicle is defined by calling STRFLT (IAERØ = 3, ITHR = 5) and then changing to BRLRL1 (barrel roll). In the argument listing for BRLRL1, the number of rolls (1), the g's to be pulled (4), and the roll rate (60 deg/sec) are specified. Flag IRØLL(3) = 2 indicates that the number of rolls required is completed. This is used as the criterion in PØLICY for switching to subroutine RTRN5, a 4-g diving turn with a 130-deg roll. See Appendix D for further instructions on the calling of individual maneuvers.

```
C
C SAMPLE POLICY FOR FLYING AIRCRAFT MANUEVERS, NO MISSILE
C
C**** SPECIFY OUTPUT
      NPRINT=2
      IPRINT(1)=1
      IPRINT(2)=3
C
C ********** FIGHTER COMMANDS **********
      GO TO (110,120,130,140),JPOL
  110 CONTINUE
      IF (TIME .G   0.1) GO TO 120
      CALL STRFLT(1,3,5)          *FIGHTER FLIES STRAIGHT FLIGHT FOR .1
      GO TO 190                    SEC, ZERO THRUST
  120 CONTINUE
      IF (V(1,6) .GT. 30.0*RAD) GO TO 130
      CALL CLIMB1(1,4.0,3,5)      *FIGHTER PULLS 4 G CLIMB UNTIL IT HAS
      JPOL=2                       CLIMBED 30 DEG, AT WHICH TIME IT
      GO TO 190                    BEGINS LEVELING OFF
  130 CONTINUE
      IF (LEVEL(1) .EQ. 2) GO TO 140
      CALL STRLVL(1,3,5,LEVEL)    *WHEN LEVEL(I)=2, VEHICLE HAS LEVELED
      JPOL=3                       OFF TO HORIZONTAL (V(I,6)=0) AND
      GO TO 190                    FLIES STRAIGHT
  140 CONTINUE
      CALL STRFLT(1,3,5)
      JPOL=4
  190 CONTINUE
C
C ********** MISSILE COMMANDS **********
      GO TO (210,220,230),KPOL
  210 CONTINUE
      CALL CAPFLT(2,2)        *MISSILE NOT BEING USED SO QUANTITIES
      GO TO 290                ZEROED OUT
  220 CONTINUE
  230 CONTINUE
  290 CONTINUE
C
C ********** TARGET COMMANDS **********
      GO TO (310,320,330),LPOL
  310 CONTINUE
      IF (TIME .GE. 1.0) GO TO 320    *TARGET FLIES STRAIGHT FLIGHT FOR
      CALL STRFLT(3,3,5)               1 SEC
      GO TO 390
  320 CONTINUE                                  *TARGET THEN MAKES ONE
      CALL BRLRL1(3,1.0,IROLL,4.0,60.0,3,5)      BARREL ROLL (4G, ROLL
      IF (IROLL(3) .EQ. 2) GO TO 330             60 DEG/SEC). IROLL(I)=2
      LPOL=2                                     INDICATES ROLLS ASKED
      GO TO 390                                  FOR ARE COMPLETED
  330 CONTINUE
      CALL RTRN5(3,4.0,130.0,3,5)     *TARGET PULLS 4 G DIVING TURN
      LPOL=3                           WITH 130 DEG ROLL
  390 CONTINUE
C
      RETURN
      END
```

Fig. 15 — Sample PØLICY subroutine for aircraft maneuvers, no missiles

Initial Data

After the PØLICY subroutine is written, the input data for setting up the initial conditions of the problem must be read in. The initial position and velocity of the vehicles must be specified. In this example, vehicle 1 is situated at the origin at an altitude of 15,000 ft. Vehicle 3 is 8000 ft down the y-axis at an altitude of 20,000 ft. The velocity of vehicle 1 is 1340 ft/sec; it is headed down the -x-axis (V(1,5) = 180.0 deg). Vehicle 2 has a velocity of 1000 ft/sec, and it is headed down the y-axis (V(3,5) = 90 deg). See Fig. 16. Other data used in this example are the following:

Starting value for integration step size (DTØ) = 0.01.
Structural lateral acceleration limit of vehicles = 7.5, 7.5.
Area of vehicles (AREA) = 196.0, 385.0.
Initial weight of vehicles (WO) = 16,669.0, 33,283.0.
Time interval for printing output (DTPØ) = 0.5.
Time value at which program is to stop (TØTAL) = 25.0.

Figure 17 shows how the data are entered on the input form.

EXAMPLE 2: LAUNCHING ONE MISSILE

The first vehicle (fighter) is to pursue the third vehicle (target) until a launching position is reached. At this time, the missile is launched from the fighter and the program is to find the closest miss distance of the missile and then terminate. Aircraft tables are to be used to simulate the aerodynamics of vehicles 1 and 3.

PØLICY Subroutine

The following PØLICY statements define the problem:

Vehicle 1 (F-104 Intercepting Aircraft)

- o Fly lead collision navigation course; military thrust.
- o If the range to target is less than 6500 ft and the angle off the target's tail is less than 30 deg, launch missile.

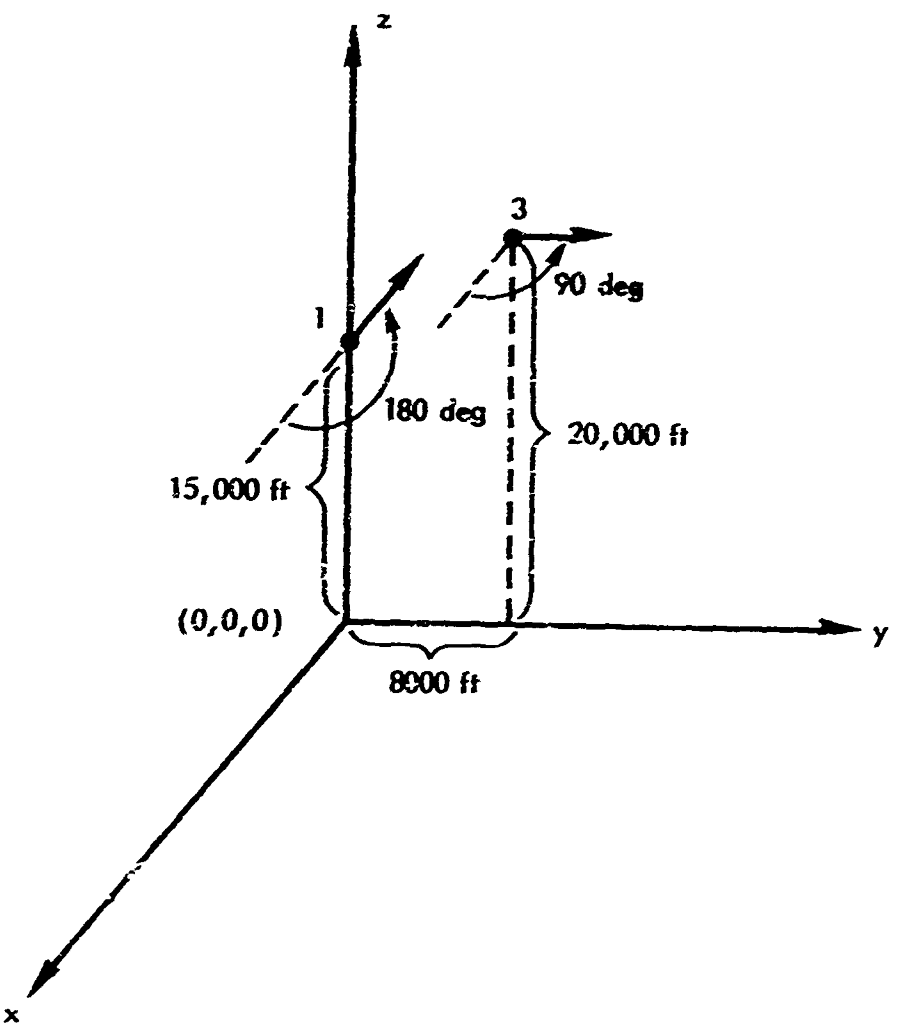


Fig. 16 — Three-dimensional diagram of initial conditions for Example 1

TACTICS PROGRAM - INPUT FORM[a]

VEHICLE TABLE FLAGS (0 OR 1)[b]

6	12	18
#1	#2	#3
0	0	0

TITLE CARD[b]

5 10 15 20 25 30 35 40 45 50 55 60 65 70 72

TACTICS - SAMPLE POLICY FOR AIRCRAFT MANUEVERS, NO MISSILE

MAIN SET OF INITIAL CONDITIONS

Columns: 1 4 6 15 18 20 29 32 34 43 46 48 57 60 62 71 74

Vehicle #1

Card					
0001	Cartesian (0)	x	y	+1.5 z E+05	+1.6699 Initial weight E+05
0006	Spherical (1)	r	θ	φ	+1.96 Reference area E+03
0011	Cartesian (0)	ẋ	ẏ	ż	+7.5 $a_{S_{max}}$ (lateral g's) E+01
0016	+1 Spherical (1 or 2) E+01	+1320 V E+04	+180 θ_V E+03	γ	Mach. no. flag

Vehicle #2

Card					
0025	Initial weight	Reference area	$a_{S_{max}}$ (lateral g's)	+2 Captive flight (0,1,2,3) E+01	Maximum range

Vehicle #3

Card					
0040	Cartesian (0)	x	+8 y E+04	+2 z E+05	+33283 Initial weight E+05
0045	Spherical (1)	r	θ	φ	+385 Reference area E+03
0050	Cartesian (0)	ẋ	ẏ	ż	+7.5 $a_{S_{max}}$ (lateral g's) E+01
0055	+1 Spherical (1 or 2) E+01	+1 V E+04	+9 θ_V E+02	γ	

PRINT

Card					
0062	Initial time	+5 Print interval E+00	+25 Termination time E+02		

GUIDANCE (3 VEHICLES)

Card	Number of time constants	τ(1)	τ(2)	τ(3)	λ navigation constant
0065					
0070					
0075					

AERODYNAMICS AND PROPULSION (3 VEHICLES)

Card	$C_{L_{max}}$	C_{D0}	dC_D/dC_L^2	$dC_L/d\alpha$	α_0
0080					
0085					
0090					

Card	Military thrust	Afterburner thrust	Specific impulse	Burnout weight	Boost acceleration
0095					
0100					
0105					

ROUND-EARTH OPTION[c]

Card					
0110	Round-earth flag	Altitude (#1)	Longitude (#1)	Latitude (#1)	Velocity flag (#1)
0115	Rotation (1)	Altitude (#3)	Longitude (#3)	Latitude (#3)	Velocity flag (#3)
0120	Origin longitude	Origin latitude			

MISCELLANEOUS OPTIONS AND EXTRA INPUTS[c]

Card					
0122	Integration flag	No. of significant digits req.	Miss calculation R_{min}	θ_V aiming error (#1)	θ_V aiming error (#2)
0127	θ_V aiming error (#3)	γ aiming error (#1)	γ aiming error (#2)	γ aiming error (#3)	Ballistic coefficient (#1)
0132	Ballistic coefficient (#2)	Ballistic coefficient (#3)			+1 E-01
[d] -					

[a]The following units are used: distance (ft), time (sec), velocity (ft/sec or Mach no.), acceleration (g's), angles (deg), weight (lb), area (ft^2)

[b]This card required.

[c]Leave this section blank if option not used.

[d]Last data card must have a minus sign in column 1.

Fig. 17 — Input data for Example 1

- o Pull constant-Mach-number, constant-altitude right turn; thrust, afterburner.

Vehicle 2 (Proportional Navigation Missile)

- o Fly captive flight until launch criterion is satisfied.
- o Launch, boost, fly unguided and then guided in accordance with guidance and aerodynamic characteristics specified in special missile subroutine.
- o When range rate (missile-target) becomes greater than zero, initiate process for finding miss distance and end program.

Vehicle 3 (F-105 Target Aircraft)

- o Fly straight and level; thrust, 80-percent throttle setting, military power.
- o When missile is launched, perform 5.5-g constant-Mach-number left turn; thrust, afterburner.

See Fig. 18 for the actual PØLICY subroutine. The output specified is standard output, aerodynamics, and attitude-angle sections.

LEADCL (lead collision) is called for the fighter, with IAERØ = 2, indicating tables for aerodynamic functions, and ITHR = 2, indicating tables for military thrust. ILAUN = 3 is a flag indicating that the missile has been launched; it is used in this case as the criterion for the fighter switching to RTRN2 (constant-Mach-number, constant-altitude right turn); ITHR = 1 indicates tables for afterburner thrust.

The missile is being held in CAPFLT(2,1), captive flight on the fighter, until the range between fighter and target (RREL(2,4)) is less than 6500 ft and the bearing angle between fighter and target is less than 30 deg. The missile is launched with boost (DELV) equal to zero, no aerodynamics (IAERØ = 3), and zero thrust (ITHR = 5). The MISILX routine is called to simulate the aerodynamics and flight of a proportional navigation missile.

The program automatically computes the closest miss distance if the missile comes within MINMR (DATA 124) of the target. This data number should be set at a large value (e.g., 1000 ft). The program will terminate after finding the miss distance unless otherwise specified.

```
C SAMPLE POLICY FOR LAUNCHING ONE MISSILE
C
C
C**** SPECIFY OUTPUT
      NPRINT=3
      IPRINT(1)=1
      IPRINT(2)=2
      IPRINT(3)=3
C
C
C ********** FIGHTER COMMANDS **********
      GO TO (110,120,130),JPOL
  110 CONTINUE
      IF (ILAUN .EQ. 3) GO TO 120      *ILAUN=3 AT LAUNCH
      CALL LEADCL(1,0.0,6500.0,2,2)
      GO TO 190                 *FIGHTER FLYING LEAD COLLISION NAVIGATION,
  120 CONTINUE                   MILITARY THRUST, UNTIL LAUNCH - THEN
      CALL RTRN2(1,2,1)          SWITCHES TO CONSTANT MACH RIGHT TURN,
      JPOL=2                     AFTER-BURNER THRUST
  130 CONTINUE
  190 CONTINUE
C
C
C ********** MISSILE COMMANDS **********
      GO TO (210,220,230),KPOL
  210 CONTINUE
      IF(ABS(BEARNG(2)) .LT. 30.0*RAD .AND. RREL(2,4) .LT. 6500.0)
     1          GO TO 220
      CALL CAPFLT(2,1)          *MISSILE LAUNCHED IF RELATIVE RANGE BETWEEN
      GO TO 290                  FIGHTER-TARGET IS LESS THAN 6500 FT AND
  220 CONTINUE                   FIGHTER IS WITHIN 30 DEG OFF TARGETS TAIL
      CALL LAUNCH(2,0.0,3,5)
  230 CONTINUE
      CALL MISILX(2)
      KPOL=3
  290 CONTINUE
C
C
C ********** TARGET COMMANDS **********
      GO TO (310,320,330),LPOL
  310 CONTINUE
      IF (ILAUN .EQ. 3) GO TO 320
      THROTL(3)=.8              *THROTL PROPORTIONS THRUST
      CALL STRFLT(3,2,2)
      GO TO 390                    *TARGET ON STRAIGHT FLIGHT UNTIL LAUNCH,
  320 CONTINUE                      THEN PULLS 5.5 G CONSTANT MACH LEFT
      CALL LTRN3(3,5.5,2,1)         TURN, AFTER-BURNER THRUST
      LPOL=2
  330 CONTINUE
  390 CONTINUE
C
C
      RETURN
      END
```

Fig. 18 — Sample PØLICY subroutine for launching one missile

Target commands are the following: Call STRFLT(IAERØ = 2, ITHR = 2) until the missile is launched (ILAUN = 3). THRØTL(3) is set equal to 0.8 to give an 80-percent throttle setting. The target then pulls LTRN3 (constant-Mach-number left turn), where the g's are specified as 5.5 in the argument. ITHR = 1 indicates that tables are to be used for obtaining a value for afterburner thrust. See Appendix D for further instructions on the calling of individual maneuvers.

Initial Data

Since table values are used for the aerodynamic and propulsion computations, data decks for the F-104 and F-105 aircraft are entered following the JVEH flag card, which indicates which vehicles have tables.

Figure 19 shows the initial position and velocity of the vehicles as entered in the data. The fighter is at the origin at an altitude of 20,000 ft. The target is 5000 ft down the -x-axis, 9000 ft along the y-axis, and at 25,000 ft altitude. In this case, velocity is read in as Mach number instead of as ft/sec. To do this, JATMØS (DATA 20) is set equal to 1. Only the magnitude of the fighter's velocity (i.e., Mach 1.2) is read in, and the program automatically aims the fighter at the target for a lead collision course. The aiming routine is triggered by setting IVF (DATA 16) equal to 2. (See Appendix E for details on the flags for reading position and velocity.) The target is flying Mach 0.9, heading down the y-axis (V(3,5) = 90 deg). Other data used in this example are the following:

Starting value for integration step size (DTØ) = 0.01.

Structural lateral acceleration limit of aircraft (ASMAX) = 7.3, 8.0.

Navigation constant for closed-loop guidance routines (LAMDAO) = 4.0, 4.0.

Area of aircraft (AREA) = 196.0, 385.0.

Initial weight of aircraft (WO) = 16,699.0, 33,287.0.

Time interval for printing output (DTPØ) = 0.5.

Time value at which program is to stop (TØTAL) = 25.0.

Flag specifying that initial-condition value of velocity of aircraft is expressed as Mach number (JATMØS) = 1.

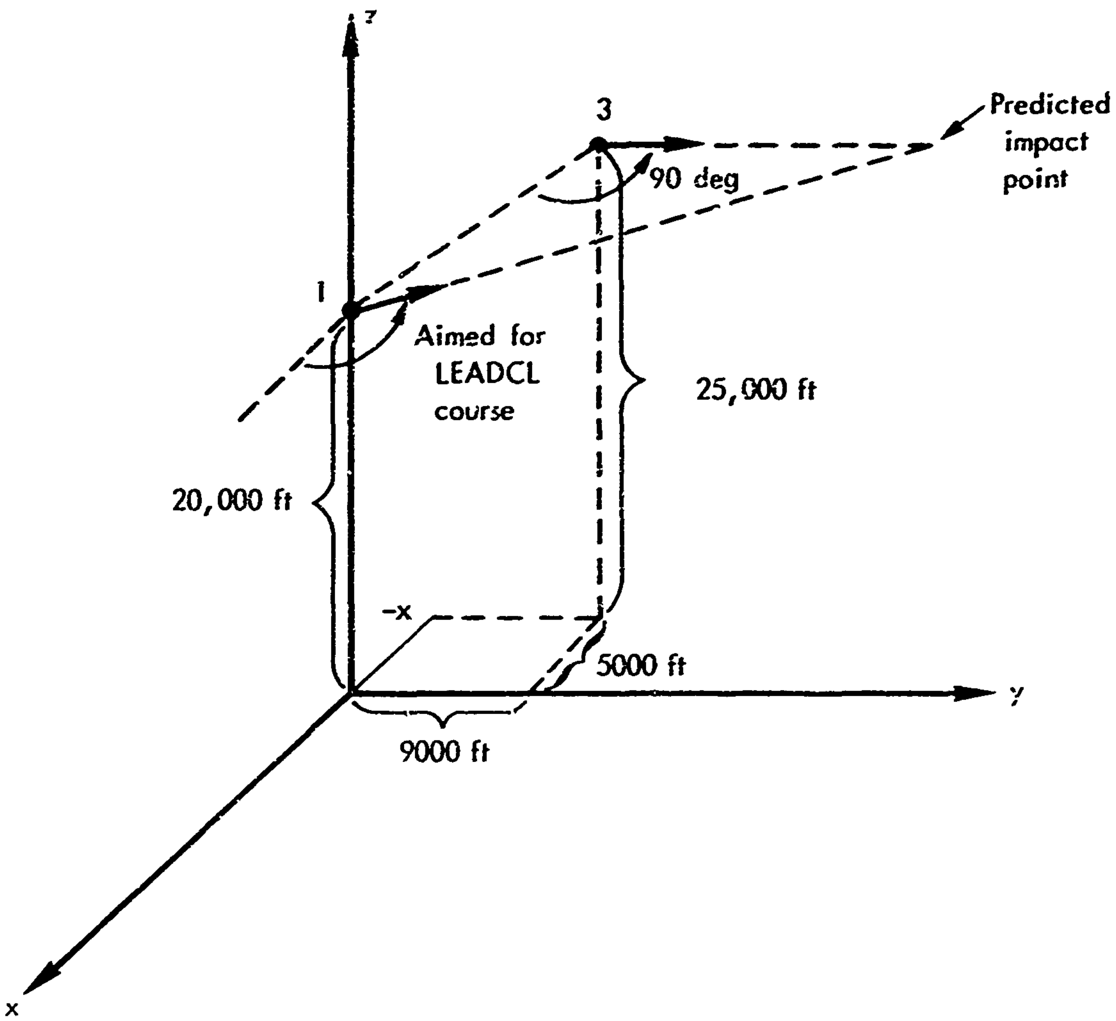


Fig. 19 — Three-dimensional diagram of initial conditions for Example 2

Missile range to target within which program will automatically initiate process for miss-distance computation (MINMR) = 1000.0 ft.

Since missile parameters such as weight and area are defined within the missile subroutine, their values do not have to be read in. The initial position and velocity of the missile are set equal to that of the fighter.

Figure 20 shows the initial data for this example entered on the input form. Refer to Section X for instructions on preparing such a data package.

EXAMPLE 3: GROUND-LAUNCHED MISSILE

A SAM is launched from the ground at a constant-velocity target. Analytic functions are used to compute the missile's aerodynamics. The closest miss distance between missile and target is to be found and the program terminated.

PØLICY Subroutine

The following PØLICY statements define the problem:

Vehicle 1

- Not used.

Vehicle 2 (SAM)

- Launch missile at time zero with a constant thrust of 5397 lb and an initial velocity of 100 ft/sec.
- Fly in 0-g dive until burnout at 1.0 sec.
- Continue 0-g dive until time to guide; thrust: 0.0.
- Begin guidance 1.0 sec after launch, fly proportional navigation with a time lag of 0.2 sec; thrust: 0.0.

Vehicle 3 (Target Aircraft)

- Fly straight and level at constant speed, no aerodynamic or propulsion computations.

TACTICS PROGRAM - INPUT FORM[a]

VEHICLE TABLE FLAGS (0 OR 1)[b]

#1 (col. 6)	#2 (col. 12)	#3 (col. 18)
1	0	1

TITLE CARD[b]

Columns 5 10 15 20 25 30 35 40 45 50 55 60 65 70 72

TACTICS - SAMPLE POLICY FOR LAUNCHING ONE MISSILE

MAIN SET OF INITIAL CONDITIONS

Columns: 1 4 6 15 18 20 29 32 34 43 46 48 57 60 62 71 74

Vehicle #1

Card					
0001	Cartesian (0)	x	y	z: +2 E+05	Initial weight: +16699 E+05
0006	Spherical (1)	r	θ	φ	Reference area: +196 E+03
0011	Cartesian (0)	ẋ	ẏ	ż	$a_{S_{max}}$ (lateral g's): +13 E+01
0016	Spherical (1 or 2): +2 E+01	V: +12 E+01	θ_V	γ	Mach no. flag: +1 E+01

Vehicle #2

Card					
0025	Initial weight	Reference area	$a_{S_{max}}$ (lateral g's)	Captive flight (0,1,2,3): +1 E+01	Maximum range

Vehicle #3

Card					
0040	Cartesian (0)	x: -5 E+04	y: +9 E+04	z: +25 E+05	Initial weight: +33287 E+05
0045	Spherical (1)	r	θ	φ	Reference area: +385 E+03
0050	Cartesian (0)	ẋ	ẏ	ż	$a_{S_{max}}$ (lateral g's): +8 E+01
0055	Spherical (1 or 2): +1 E+01	V: +9 E+00	θ_V: +9 E+02	γ	

PRINT

Card					
0062	Initial time	Print interval: +5 E+00	Termination time: +25 E+02		

GUIDANCE (3 VEHICLES)

Card	Number of time constants	τ (1)	τ (2)	τ (3)	λ navigation constant
0065					+4 E+01
0070					
0075					+4 E+01

AERODYNAMICS AND PROPULSION (3 VEHICLES)

Card	$C_{L_{max}}$	C_{D0}	dC_D/dC_L^2	$dC_L/d\alpha$	α_0
0080					
0085					
0090					

Card	Military thrust	Afterburner thrust	Specific impulse	Burnout weight	Boost acceleration
0095					
0100					
0105					

ROUND-EARTH OPTION[c]

Card					
0110	Round-earth flag	Altitude (#1)	Longitude (#1)	Latitude (#1)	Velocity flag (#1)
0115	Rotation (1)	Altitude (#3)	Longitude (#3)	Latitude (#3)	Velocity flag (#3)
0120	Origin longitude	Origin latitude			

MISCELLANEOUS OPTIONS AND EXTRA INPUTS[c]

Card					
0122	Integration flag	No. of significant digits req.	Miss calculation R_{min}: +1 E+04	θ_V aiming error (#1)	θ_V aiming error (#2)
0127	θ_V aiming error (#3)	γ aiming error (#1)	γ aiming error (#2)	γ aiming error (#3)	Ballistic coefficient (#1)
0132	Ballistic coefficient (#2)	Ballistic coefficient (#3)			+1 E-01
[d]					

[a] The following units are used: distance (ft), time (sec), velocity (ft/sec or Mach no.), acceleration (g's), angles (deg), weight (lb), area (ft^2)

[b] This card required.

[c] Leave this section blank if option not used.

[d] Last data card must have a minus sign in column 1

Fig. 20 — Input data for Example 2

Figure 21 shows the PØLICY subroutine. The printout is to consist of standard output, aerodynamics, and attitude angles. Because the fighter is not used in this run, its quantities are set to zero by calling CAPFLT(1,2).

The missile is launched at time zero; DIVE1 with 0 g's is called (i.e., a ballistic trajectory). Analytic functions are used to compute aerodynamics, so IAERØ = 1; constant thrust is indicated, so ITHR = 4. TLAUN(2) is the time of launch and can be used as a criterion in PØLICY. Here it is used both to determine burnout (TBURN(1)) and the time the missile is to begin guiding (TGUIDE(2)). Missile guidance begins by calling PRØNAV (proportional navigation). Thrust now equals zero, so ITHR = 5. A time lag is added to the missile at this point by setting ITAU(2) = 1, which indicates that only one time lag is being used; TAU(2,1) = 0.2, indicating a 0.2-sec lag. STRFLT is called for the target with no aerodynamic or thrust computations, so IAERØ = 3 and ITHR = 5. See Appendix D for further instructions on the calling of individual maneuvers.

Initial Data

Figure 22 is a three-dimensional diagram of the initial positions and velocities of the missile and target specified in the input data. The program is capable of reading in values for the position and velocity of the missile, but in this case the initial conditions are set to those of the fighter instead, i.e., the missile's values are read in as the fighter's position and velocity. The fighter values are later zeroed out, since they are not used in this problem. This seemingly needless complexity may be used to advantage for automatic aiming of vehicle 1 against 3 or vice versa using the flags IVF = 2 or IVT = 2. This will automatically call the AIM routine and, in this case, provide for initially aiming 2 against 3 (see the description of subroutine AIM in Appendix C).

The missile is initially placed at the origin, and the altitude component is accordingly zero. The target begins its flight 3000 ft down the -x-axis, 10,000 ft along the y-axis, and at a 500-ft altitude.

```
C SAMPLE POLICY FOR GROUND LAUNCHED MISSILE
C
C**** SPECIFY OUTPUT
      NPRINT=3
      IPRINT(1)=1
      IPRINT(2)=2
      IPRINT(3)=3
C
C
C ********** FIGHTER COMMANDS **********
      GO TO (110,120,130),JPOL
  110 CONTINUE
      CALL CAPFLT(1,2)        *FIGHTER NOT BEING USED SO QUANTITIES
      GO TO 190                ZEROED OUT
  120 CONTINUE
  130 CONTINUE
  190 CONTINUE
C
C
C ********** MISSILE COMMANDS **********
      GO TO (210,220,230,240,250),KPOL
  210 CONTINUE
      CALL LAUNCH(2,0.0,3,5)        *MISSILE LAUNCHED AT TIME ZERO
  220 CONTINUE
      IF ((TIME-TLAUN(2)) .GT. TBURN1) GO TO 230
      CALL DIVE1(2,0.0,1,4)         *MISSILE FLIES IN ZERO G DIVE WITH
      KPOL=2                         THRUST=CONSTANT UNTIL BURNOUT(TBURN1)
      GO TO 290                      ,AT WHICH TIME THRUST SET TO ZERO
  230 CONTINUE                       AND MISSILE CONTINUES UNGUIDED
      IF ((TIME-TLAUN(2)) .GT. TGLIDE(2)) GO TO 240
      CALL DIVE1(2,0.0,1,5)
      KPOL=3
      GO TO 290
  240 CONTINUE                  *A TIME LAG OF .2 SEC IS ADDED TO MISSILE
      ITAU(2)=1                  AFTER IT BEGINS GUIDING
      TAU(2,1)=.2
  250 CONTINUE                  *WHEN TIME SINCE LAUNCH IS GREATER THAN
      CALL PRONAV(2,1,5)         TIME THE MISSILE IS TO FLY UNGUIDED
      KPOL=5                     (TGLIDE(2)), PROPORTIONAL NAVIGATION IS
  290 CONTINUE                   CALLED
C
C
C ********** TARGET COMMANDS **********
      GO TO (310,320,330),LPOL
  310 CONTINUE
      CALL STRFLT(3,3,5)      *TARGET FLIES STRAIGHT FLIGHT
      GO TO 390
  320 CONTINUE
  330 CONTINUE
  390 CONTINUE
C
C
      RETURN
      END
```

Fig. 21 — Sample PØLICY subroutine for ground-launched missile

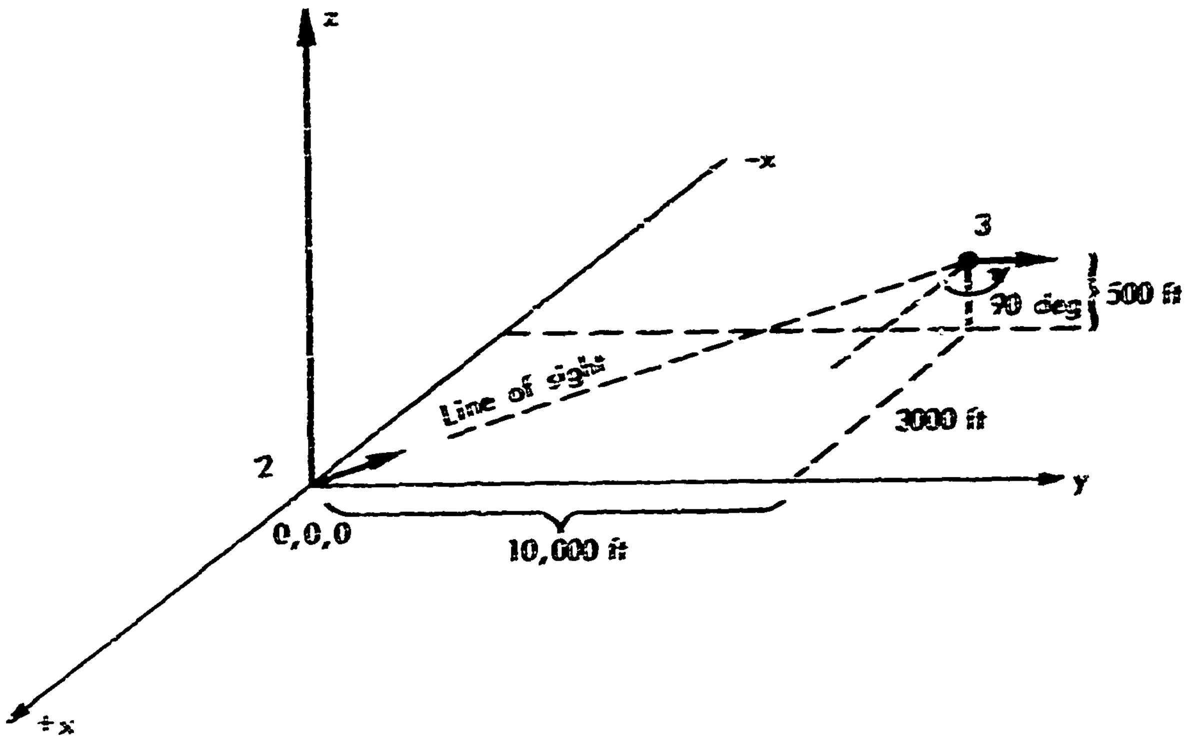


Fig. 22 — Three-dimensional diagram of initial conditions for Example 3

The missile's initial velocity is 100 ft/sec, and it is pointed directly down the LOS between missile and target, giving it a heading of 38.9 deg in the horizontal plane (V(1,5)) and 2.55 deg in the vertical plane (V(1,6)). The target has a velocity of 1340 ft/sec headed down the y-axis (V(3,6) = 90 deg).

A significant quantity of aerodynamic data must be entered for this case, since analytic functions are used to compute the missile aerodynamics, and equation constants must therefore be specified for use with Eq. (26). The following data are used in this example:

Starting value for integration step size (DTØ) = 0.01.

Time interval missile is to fly unguided after launch (TGUIDE(2)) = 1.0.

First-stage burning time of missile (TBURN1) = 1.0 sec.

Structural lateral acceleration limit of missile and target (ASMAX) = 15.0, 7.3 g's.

Constants to be used for thrust (THCØN(2)) = 5397.

Navigation constant for closed-loop guidance routines of missile (LAMDA0(2)) = 4.0.

Maximum aerodynamic lift coefficient of missile (CLMAX(2)) = 1.4.

$d(C_D)/d(C_L^2)$, constant of missile (BCØN(2)) = 0.0041666667.

$dC_L/d\alpha$ (assumed to be a constant) SLØPE(2) = 0.1.

Specific impulse of rocket motor (IMPLSE(2))* = 250.0.

Initial weight of missile (W0)* = 50.0.

Area of missile (AREA) = 0.264.

Integration mode (in this case, fixed-step Adams-Moulton) (JINTEG) = 2.

Time interval for printing output (DTPØ) = 0.5.

Time value at which program is to stop (TØTAL) = 25.0.

Missile range to target within which program will automatically initiate process for miss-distance computation (MINMR) = 1000.0.

Figure 23 shows the initial data for this example entered on the input form. Refer to Section X for instructions on preparing such a data package.

---

*These quantities must be set for computing missile weight during the propulsion interval.

TACTICS PROGRAM - INPUT FORM[a]

[illegible][b]

#1	#2	#3
[illegible]	[illegible]	[illegible]

TITLE CARD[b]

TACTICS - SAMPLE POLICY FOR LAUNCHING MISSILE FROM GROUND

[illegible] SET OF INITIAL CONDITIONS

Vehicle #1

Card					
[illegible]	Common (C)	x	y	z	Initial weight
[illegible]	[illegible]	α	θ	φ	Reference area
[illegible]	Common (C)	ẋ	ẏ	ż	g_{max} (lateral g's)
[illegible]	[illegible] E+01	+1 E+03	+989 E+02	+255 E+01	Mach no. flag

Vehicle #2

Card	Initial weight	Reference area	g_{max} (lateral g's)	Capture flight (0,1,2,3)	Maximum range
[illegible]	+5 E+02	+264 E+00	+1.5 E+02		

Vehicle #3

Card					
[illegible]	Common (C)	-3 E+04	+1.0 E+05	+5 E+03	Initial weight
[illegible]	Spherical (B)	α	θ	φ	Reference area
[illegible]	Common (C)	ẋ	ẏ	ż	+7.3 E+01 (g_{max} (lateral g's))
[illegible]	[illegible] E+01	+1,920 E+04	+9 E+02	ṙ	

PRINT

Card	Initial time	Print interval	Termination time		
[illegible]		+5 E+00	+12 E+02		

GUIDANCE (3 VEHICLES)

Card	Number of time constants	τ (1)	τ (2)	τ (3)	λ navigation constant
[illegible]					
[illegible]					+4 E+02
[illegible]					

AERODYNAMICS AND PROPULSION (3 VEHICLES)

Card	$C_{L_{max}}$	C_{D0}	$\alpha C_D/\alpha C_L^2$	$\alpha C_L/d\alpha$	a_0
0080					
0085	+1.4 E+01	+1.2 E-01	+4.1666667 E-02	+1. E+00	
0090					

Card	Military thrust	Afterburner thrust	Specific impulse	Burnout weight	Boost acceleration
0095					
0100	+5397 E+04		+250 E+03		
0105					

ROUND-EARTH OPTION[c]

Card					
0110	Round-earth flag	Altitude (#1)	Longitude (#1)	Latitude (#1)	Velocity flag (#1)
0115	Rotation (1)	Altitude (#3)	Longitude (#3)	Latitude (#3)	Velocity flag (#3)
0120	Origin longitude	Origin latitude			

MISCELLANEOUS OPTIONS AND EXTRA INPUTS[c]

Card					
0122	Integration flag: +2 E+01	No. of significant digits req.	Miss calculation R_{min}: +1 E+01	θ_V aiming error (#1)	θ_V aiming error (#2)
0127	θ_V aiming error (#3)	γ aiming error (#1)	γ aiming error (#2)	γ aiming error (#3)	Ballistic coefficient (#1)
0132	Ballistic coefficient (#2)	Ballistic coefficient (#3)			+1 E-01
[d] -[illegible]35		+1 E+01			+1 E+01

[a]The following units are used: distance (ft), time (sec), velocity (ft/sec or Mach no.), acceleration (g's), angles (deg), weight (lb), area (ft^2)

[b]This card required.

[c]Leave this section blank if option not used.

[d]Last data card must have a minus sign in column 1.

Fig. 23 — Input data for Example 3

EXAMPLE 4: LAUNCHING TWO MISSILES, RECALL

The recall features of the program provide the capability of launching more than one missile. After the missile has been launched the first time and the miss distance determined, the program continues instead of terminating, and the missile can be recalled either to vehicle 1 or to vehicle 3. The aerial combat is then continued and the missile is launched again. Therefore, if vehicle 1 misses on the first launching, it can repeat the process by launching another missile, or vehicle 3 can retaliate by launching a missile at vehicle 1.

In this example, vehicle 1 is pursuing vehicle 3 and launches a missile when the launch criteria have been satisfied. After the missile has missed the target it is recalled to vehicle 3, and now vehicle 1 is the target. The missile is now launched against vehicle 1.

PØLICY Subroutine

The following PØLICY statements define the problem:

Vehicle 1 (F-104 Intercepting Aircraft)

- Fly pursuit-course navigation; thrust, military power.
- If the range to target is less than 7000 ft, launch missile.
- After launch perform constant-Mach-number, constant-altitude left turn; thrust, afterburner.
- When heading angle in horizontal plane is greater than 270 deg, fly straight and level; thrust, military.

Vehicle 2 (First Missile)

- Hold in captive flight by fighter until launch criterion is satisfied.
- Launch, boost, fly unguided and then guided in accordance with guidance and aerodynamic characteristics specified in special missile routine.
- When range rate (missile-target) becomes greater than zero, initiate process for finding miss distance.

Vehicle 3 (F-105 Target Aircraft)

- o Fly straight and level; thrust, military.
- o When missile is launched, perform 5-g left turn; thrust, military.
- o After fighter-launched missile has missed, switch to pursuit-course navigation tracking vehicle 1; thrust, afterburner.
- o If range to fighter is less than 10,000 ft and greater than 7000 ft and the target is within 30 deg off fighter's tail, launch second missile, which has been recalled to vehicle 3.

Vehicle 4 (Second Missile)

- o Missile recalled to vehicle 3 after first missile has missed.
- o Hold in captive flight by target until launch criterion is satisfied.
- o Launch, boost, fly unguided and then guided in accordance with guidance and aerodynamic characteristics specified in special missile subroutine.
- o When range rate (missile-fighter) becomes greater than zero, initiate process for finding miss distance and terminate program.

Figure 24 shows the PØLICY subroutine. The printout is to consist of standard output, aerodynamics, and attitude angles.

Fighter flies PRSUIT (pursuit-course navigation) with table values for aerodynamics (IAERØ = 2) and table values for military thrust (ITHR = 2). At launch time (ILAUN = 3 indicates that missile has been launched) LTRN2 (constant-Mach-number, constant-altitude left turn) is called with afterburner thrust (ITHR = 1). Fighter flies LTRN2 until its horizontal heading (V(1,5)) is greater than 270 deg. This is used as the criterion for switching to STRFLT.

The first missile is held in CAPFLT(2,1), captive flight by fighter, until the range between fighter and target (RREL(2,4)) is less than 7000 ft. The missile is launched, and the MISIL1 routine is called to simulate the aerodynamics and flight of a proportional navigation missile.

To initiate the recall feature of the program, it is necessary to set the flag IMISS = 1, which causes the program to continue after finding the first missile miss. It is reset to zero for the second

```
C SAMPLE POLICY FOR LAUNCHING TWO MISSILES
C
C
C**** SPECIFY OUTPUT
      NPRINT=3
      IPRINT(1)=1
      IPRINT(2)=2
      IPRINT(3)=3
C
C
C ********** FIGHTER COMMANDS **********
      GO TO (110,120,130),JPOL
  110 CONTINUE
      IF (ILAUN .EQ. 3) GO TO 120       *ILAUN=3 AT LAUNCH
      CALL PRSUIT(1,2,2)      *FIGHTER FLIES PRSUIT COURSE NAVIGATION, MILITARY
      GO TO 190                THRUST, UNTIL MISSILE LAUNCHED
  120 CONTINUE
      IF (V(1,5) .GT. 270.0*RAD) GO TO 130
      CALL LTRN2(1,2,1)       *AT LAUNCH FIGHTER SWITCHES TO CONSTANT MACH,
      JPOL=2                   CONSTANT ALTITUDE LEFT TURN, AFTER-BURNER THRUST
      GO TO 190
  130 CONTINUE
      CALL STRFLT(1,2,2)      *WHEN HEADING ANGLE IN HORIZONTAL PLANE (THETA)
      JPOL=3                   IS GREATER THAN 270 DEG, FIGHTER FLIES STRAIGHT
  190 CONTINUE                 FLIGHT, MILITARY THRUST
C
C
C ********** MISSILE COMMANDS **********
      GO TO (210,220,230),KPOL
  210 CONTINUE
      IF (KMISS .EQ. 1) GO TO 290
      IMISS=1                 *FLAG FOR PROGRAM TO CONTINUE AFTER MISSILE MISS
      IF (RREL(2,4) .LE. 7000.0) GO TO 220
      CALL CAPFLT(2,1)        *IF RELATIVE RANGE BETWEEN FIGHTER-TARGET IS LESS
      GO TO 290                THAN 7000 FT., LAUNCH MISSILE.  OTHERWISE MISSILE
  220 CONTINUE                 HELD IN CAPTIVE FLIGHT BY FIGHTER
      CALL LAUNCH(2,0.0,3,5)
  230 CONTINUE
      IF (IMISS .EQ. 2) GO TO 240
      CALL MISIL1(2)          *IF IMISS=2 PROGRAM IS READY TO CONTINUE AFTER
      KPOL=3                   FINDING MINIMUM MISS DISTANCE OF FIRST MISSILE
      GO TO 290
  240 CONTINUE
      KMISS=1                 *KMISS=1 INDICATES SECOND MISSILE IS TO BE USED
      KPOL=1
  290 CONTINUE
C
C
C ********** TARGET COMMANDS **********
```

Continued

Fig. 24 — Sample PØLICY subroutine for launching two missiles

```
      GC TC (310,32C,33C),LPCL
  31C CCNTINUE
      IF (ILALN .EC. 3) GC TO 32C
      CALL STRFLT(3,2,2)      *TARGET FLIES STRAIGHT FLIGHT, MILITARY THRUST
      GC TC 39C
  32C CCNTINUE
      IF (KMISS .EC. 1) GC TO 33C
      CALL LTRN1(3,5.C,2,2)   *AFTER LALNCH, TARGET PULLS 5 G LEFT TURN,
      LPCL=2                   MILITARY THRLST
      GC TC 39C
  33C CCNTINUE
      CALL PRSUIT(3,2,1)      *AFTER FIGHTER LAUNCHEC MISSILE HAS MISSEC, TARGET
      LPCL=3                   TARGET SWITCHES TC PURSLIT CCURSE NAVIGATICN,
  39C CCNTINLE                 AFTER-BLRNER THRLST
C
C
C ********** SECCNC MISSILE CCMMANCS **********
      GC TC (410,42C,43C,44C),MPCL
  41C CCNTINUE
      IF (KMISS .NE. 1) GC TO 49C
      IMISS=0                 *FLAG FCR PRCGRAM TO STCP AFTER FINDING MINIMUM
  42C CCNTINUE                 MISS CISTANCE FCR SECCNC MISSILE
      IF (RREL(2,4) .LT. 1CCCC.C .ANC. RREL(2,4) .GT. 7CCC.0 .ANC.
     1ABS(BEARNG(2) .GT. 15C.C*RAC) GC TC 430
      CALL CAPFLT(2,3)        *MISSILE LAUNCHEC IF RELATIVE RANGE BETWEEN TARGET
      MPCL=2                   -FIGHTER IS GREATER THAN 7OCC FT. ANC LESS THAN
      GC TC 49C                1C,CCO FT., AND THE TARGET IS WITHIN 30 CEG CFF
  43C CCNTINUE                 FIGHTERS TAIL.  OTHERWISE MISSILE IS HELC IN
      CALL LALNCH(2,0.C,3,5)   CAPTIVE FLIGHT BY TARGET
  44C CCNTINUE
      CALL MISIL1(2)
      MPCL=4
  49C CCNTINUE
C
C
      RETURN
      ENC
```

Fig. 24 (continued)

missile, and the program will terminate after the miss-distance calculation. IMISS = 2 is a flag set in the program to indicate that the miss distance for the first missile launch has been computed and that the program is ready to continue. This flag is used as the criterion in PØLICY for triggering the flag KMISS = 1, which indicates that the first missile is no longer in flight and that the second missile's commands should now be followed. (See the first and second missile command sections in PØLICY, Fig. 24.)

The second missile is held in captive flight by target CAPFLT(2,3) until the target-fighter range (RREL(2,4)) is greater than 7000 ft but less than 10,000 ft and the fighter-target bearing angle is greater than 150 deg. (See Section XI for the definition of bearing angle.) The missile is launched against the fighter and MISIL1 routine called. Flag IMISS = 0 indicates that the program will stop after determining miss distance.

Target flies STRFLT (IAERØ = 2, ITHR = 2) until the first missile is launched (ILAUN = 3). It then pulls LTRN1 (simple left turn) with 5 g's specified until the first missile has missed and is recalled to the target (this is indicated by KMISS = 1). Target now flies PRSUIT against the fighter attempting to arrive within firing range at afterburner thrust, ITHR = 1. See Appendix D for instructions on calling the optional subroutines used in PØLICY.

Initial Data

Since table values are used for computing aerodynamics, data decks for the F-104 and F-105 are entered following the JVEH flag card, as shown in Fig. 11.

Figure 25 is a three-dimensional diagram of the initial positions and velocities used in this example. As can be seen, the fighter is 6000 ft along the y-axis at 15,000 ft altitude. It has a velocity of Mach 0.92 (velocity is given in Mach number instead of ft/sec if JATMØS (DATA 20) is set equal to 1) and a horizontal heading of 210 deg (V(1,5)). The target's initial position is 8000 ft down the -x-axis and at 15,000 ft altitude. Initial velocity is Mach 0.87 with 180-deg heading angle (V(3,5)). Other data needed for this example are the following:

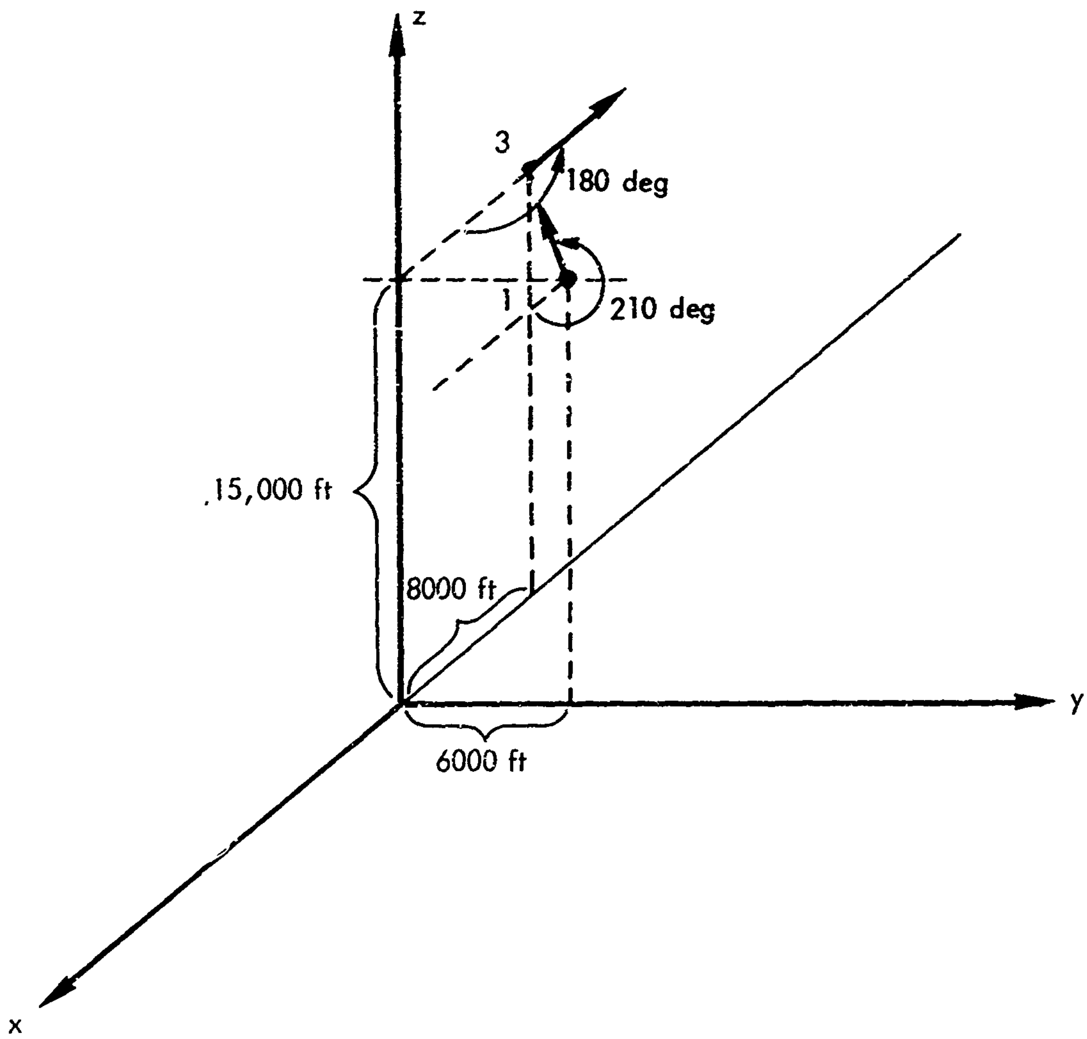


Fig. 25 — Three-dimensional diagram of initial conditions for Example 4

Starting value for integration step size (DTØ) = 0.01.

Structural lateral acceleration limit of aircraft (ASMAX) = 6.8, 7.0.

Navigation constants for closed-loop guidance routines (LAMDAO) = 40.0, 40.0.

Area of aircraft (AREA) = 196.0, 385.0.

Initial weight of aircraft (WO) = 16,699.0, 33,283.0.

Time interval for printing output (DTPØ) = 0.5.

Time value at which program is to stop (TØTAL) = 60.0.

Flag specifying that initial-condition value of velocity of aircraft is expressed as Mach number (JATMØS) = 1.

Missile range to target within which program will automatically initiate process for miss-distance computation (MINMR) = 1000.0.

Because missile constants and variables are defined within the missile subroutines, their values do not have to be read in. The initial position and velocity of the missile are set equal to those of the fighter.

Figure 26 shows the initial data for this example as entered on the input form. Refer to Section X for instructions for preparing such a data package.

EXAMPLE 5: USING RESTORE FEATURE TO FIRE 20-MM PROJECTILES

After vehicle 2 is launched and a hit or miss has occurred, this option enables the program to restore all numerical values existing at launch time. Now events may proceed, a new launch may take place, and characteristics and parameters may change. In this case the restore feature is used to fire consecutive projectiles at 0.01-sec intervals at a ground target. A projectile is launched first at a given range and the miss distance computed. The numerical values are then restored to those existing at launch time. Integration then occurs, and the next projectile may be launched at a subsequent time.

PØLICY Subroutine

The following PØLICY statements define the problem:

TACTICS PROGRAM - INPUT FORM[a]

VEHICLE TABLE FLAGS (0 OR 1)[b]

6 (#1)	12 (#2)	18 (#3)
1	0	1

TITLE CARD[b]

TACTICS - SAMPLE PØLICY FØR LAUNCHING TWØ MISSILES

MAIN SET OF INITIAL CONDITIONS

Vehicle #1

Card (1–4)	6–18	20–32	34–46	48–61	62–74
0001	Cartesian (0)	x	y: +6 E+04	z: +15 E+05	Initial weight: +16699 E+05
0006	Spherical (1)	r	θ	φ	Reference area: +196 E+03
0011	Cartesian (0)	ẋ	ẏ	ż	a_{Smax} (lateral g's): +68 E+01
0016	Spherical (1 or 2): +1 E+01	V: +92 E+00	θ_V: +210 E+03	γ	Mach no. flag: +1 E+01

Vehicle #2

Card					
0025	Initial weight	Reference area	a_{Smax} (lateral g's)	Captive flight (0,1,2,3): +1 E+01	Maximum range

Vehicle #3

Card					
0040	Cartesian (0)	x: -8 E+04	y	z: +15 E+05	Initial weight: +33283 E+05
0045	Spherical (1)	r	θ	φ	Reference area: +385 E+03
0050	Cartesian (0)	ẋ	ẏ	ż	a_{Smax} (lateral g's): +7 E+01
0055	Spherical (1 or 2): +1 E+01	V: +87 E+00	θ_V: +180 E+03	γ	

PRINT

Card	Initial time	Print interval	Termination time		
0062		+5 E+00	+6 E+02		

GUIDANCE (3 VEHICLES)

Card	Number of time constants	τ(1)	τ(2)	τ(3)	λ navigation constant
0065					+4 E+02
0070					
0075					+4 E+02

AERODYNAMICS AND PROPULSION (3 VEHICLES)

Card	C_{Lmax}	C_{D0}	dC_D/dC_L^2	$dC_L/d\alpha$	α_0
0080					
0085					
0090					

Card	Military thrust	Afterburner thrust	Specific impulse	Burnout weight	Boost acceleration
0095					
0100					
0105					

ROUND-EARTH OPTION[c]

Card					
0110	Round-earth flag	Altitude (#1)	Longitude (#1)	Latitude (#1)	Velocity flag (#1)
0115	Rotation (1)	Altitude (#3)	Longitude (#3)	Latitude (#3)	Velocity flag (#3)
0120	Origin longitude	Origin latitude			

MISCELLANEOUS OPTIONS AND EXTRA INPUTS[c]

Card					
0122	Integration flag	No. of significant digits req.	Miss calculation R_{min}: +1 E+04	θ_V aiming error (#1)	θ_V aiming error (#2)
0127	θ_V aiming error (#3)	γ aiming error (#1)	γ aiming error (#2)	γ aiming error (#3)	Ballistic coefficient (#1)
0132	Ballistic coefficient (#2)	Ballistic coefficient (#3)			+1 E-01
[d] -					

[a] The following units are used: distance (ft), time (sec), velocity (ft sec or Mach no.), acceleration (g's), angles (deg), weight (lb), area (ft^2)

[b] This card required

[c] Leave this section blank if option not used.

[d] Last data card must have a minus sign in column 1

Fig. 26 — Input data for Example 4

Vehicle 1 (F-104 Pursuing Aircraft)

- o Fly pursuit-course navigation against a stationary ground target; thrust, military.
- o When a ground range of 3000 ft has been covered, begin launching 20-mm projectiles at 0.01-sec intervals.
- o If altitude falls below 5000 ft, perform a 4-g climb; thrust, afterburner.

Vehicle 2 (20-mm Projectile)

- o Fly captive flight until launch criterion is satisfied.
- o Launch with an incremental velocity of ΔV and fly in accordance with characteristics specified in B20MM subroutine, i.e., a ballistic trajectory.
- o Find miss distance at point of ground impact, restore program to values of launch time, launch second projectile 0.01 sec later, and continue until five projectiles have been fired.

Vehicle 3 (Stationary Ground Target)

- o Remain stationary on ground.

Figure 27 shows the PØLICY subroutine. The printout is to consist of standard output, aerodynamics, and attitude angles. To use the restore feature of the program, flag ISTØRE = 1 must be set. Flag IMISS = 1 must also be specified so that the program will not stop after computing the missile miss distance. Both these flags are set the first time through PØLICY and must be reset for each separate launch.

The fighter is flying PRSUIT (pursuit-course navigation) with table values for aerodynamics (IAERØ = 2) and table values for military thrust (ITHR = 2). If its altitude falls below 5000 ft, the fighter begins a 4-g climb (CLIMB1) with afterburner thrust (ITHR = 1) in order to pull out. The projectiles are held in captive flight by the fighter (CAPFLT(2,1)), until the fighter has flown 3000 ft along the y-axis; i.e., the y-component (R(1,2)) is greater than 3000. Each projectile is launched with a boost velocity (DELV) of 2800 ft/sec, and the B20MM subroutine is called to determine the aerodynamics and flight path.

```
C  SAMPLE POLICY USING STORE TO FIRE 20MM CANNON PROJECTILES
C
C**** SPECIFY OUTPUT
      NPRINT=3
      IPRINT(1)=1
      IPRINT(2)=2
      IPRINT(3)=3
C
C ********** FIGHTER COMMANDS **********
      GO TO (110,120,130),JPOL
  110 CONTINUE
      IMISS=1                  *FLAG FOR PROGRAM TO CONTINUE AFTER MISSILE MISS
      ISTORE=1                 *FLAG FOR STORING VALUES AT LAUNCH
  120 CONTINUE
      IF (R(1,3) .LE. 5000.0) GO TO 130
      CALL PRSUIT(1,2,2)       *FIGHTER ON PRSUIT COURSE NAVIGATION, TABLE
      JPOL=2                             TABLE VALUES FOR AERODYNAMICS,
      GO TO 190                          MILITARY THRUST
  130 CONTINUE
      CALL CLIMB1(1,4.0,2,1)            *IF ALTITUDE IS LESS THAN 5000
  140 CONTINUE                           FT, SWITCH TO 4 G CLIMB TO
  190 CONTINUE                           PULL OUT, AFTER-BURNER THRUST
C ********** MISSILE COMMANDS **********
      GO TO (210,220,230,240,250),KPOL
  210 CONTINUE
      IF (R(1,2) .GE. 3000.0) GO TO 220  *IF FIGHTER HAS FLOWN 3000 FT
      CALL CAPFLT(2,1)                   ALONG Y-AXIS, LAUNCH PROJECTILE
      GO TO 290                          WITH DELV=2800. OTHERWISE HELD
  220 CONTINUE                           IN CAPTIVE FLIGHT BY FIGHTER
      CALL LAUNCH(2,2800.0,3,5)
  230 CONTINUE
      IF (ISTORE .EQ. 0) GO TO 240      *ISTORE SET BACK TO ZERO WHEN
      CALL B20MM(2)                      PROGRAM HAS RESTORED ITSELF TO
      KPOL=3                             LAUNCH VALUES AFTER COMPUTING
      GO TO 290                          THE PREVIOUS PROJECTILE MISS
  240 CONTINUE
      JINTEG=2                  *FIXED STEP INTEGRATION FOR PROJECTILES
      IMISS=1                   *IMISS RESET FOR EACH LAUNCHING
      IF (II .EQ. 3) IMISS=0    *IMISS=0, STOP AFTER 5TH PROJECTILE
      ISTORE=1                  *ISTORE RESET FOR EACH LAUNCHING
      KPOL=5
      KK=0
  250 CONTINUE
      IF (KK .NE. 0) GO TO 220  *AFTER RESTORING PROGRAM TO VALUES AT
      CALL CAPFLT(2,1)           LAUNCH TIME, INTEGRATE AHEAD .01 SEC
      KK=1                       BEFORE LAUNCHING NEXT PROJECTILE
      II=II+1
      GO TO 290
  260 CONTINUE
  290 CONTINUE
C************************TARGET COMMANDS ***********************
      GO TO (310,320,330,340),LPOL
  310 CONTINUE
      GO TO 390                         *USED AS STATIONARY GROUND TARGET
  390 CONTINUE
      RETURN
      END
```

Fig. 27 — Sample PØLICY subroutine for firing 20-mm projectiles

At ground impact, the integration routine backs up to obtain the distance between the impact point and the target.* Since IMISS = 1, the program then continues instead of ending, and the values are restored to those of launch time. At this point the flag ISTØRE = 0 indicates the program has been restored. This is used as the criterion for switching back to CAPFLT so that the program can integrate ahead 0.01 sec before launching again. This process continues until five projectiles have been launched. After the fifth launching, IMISS = 0, and the program ends after computing closest miss. Vehicle 3 is a stationary ground target and therefore receives no velocity or acceleration commands. See Appendix D for further instructions on the calling of individual maneuvers.

Initial Data

Since table values are used for computing the aerodynamics of the fighter, the F-104 data deck is entered immediately following the JVEH flag card, as shown in Fig. 11.

Figure 28 is a three-dimensional diagram describing the initial positions and velocities of the vehicles in this example. The fighter is located at the origin at an altitude of 7000 ft. It is headed directly down the y-axis (V(1,5) = 90 deg) with a velocity of 1000 ft/sec. The target is located on the ground 6000 ft along the y-axis. Since it is stationary, it has no initial velocity. Other data needed for this example are the following:

Starting value for integration step size (DTØ) = 0.01.

Structural lateral acceleration limit of fighter (ASMAX) = 7.3.

Navigation constant for closed-loop guidance routine (LAMDAO) = 40.0.

Area of fighter (AREA) = 196.1.

Initial weight of fighter (WO) = 19,470.0.

---

*This is *not* the same as the point of closest approach or miss distance as usually defined. (See Section XIII.) TACTICS automatically makes the distinction if the altitude of the target R(3,3) is *exactly* zero.

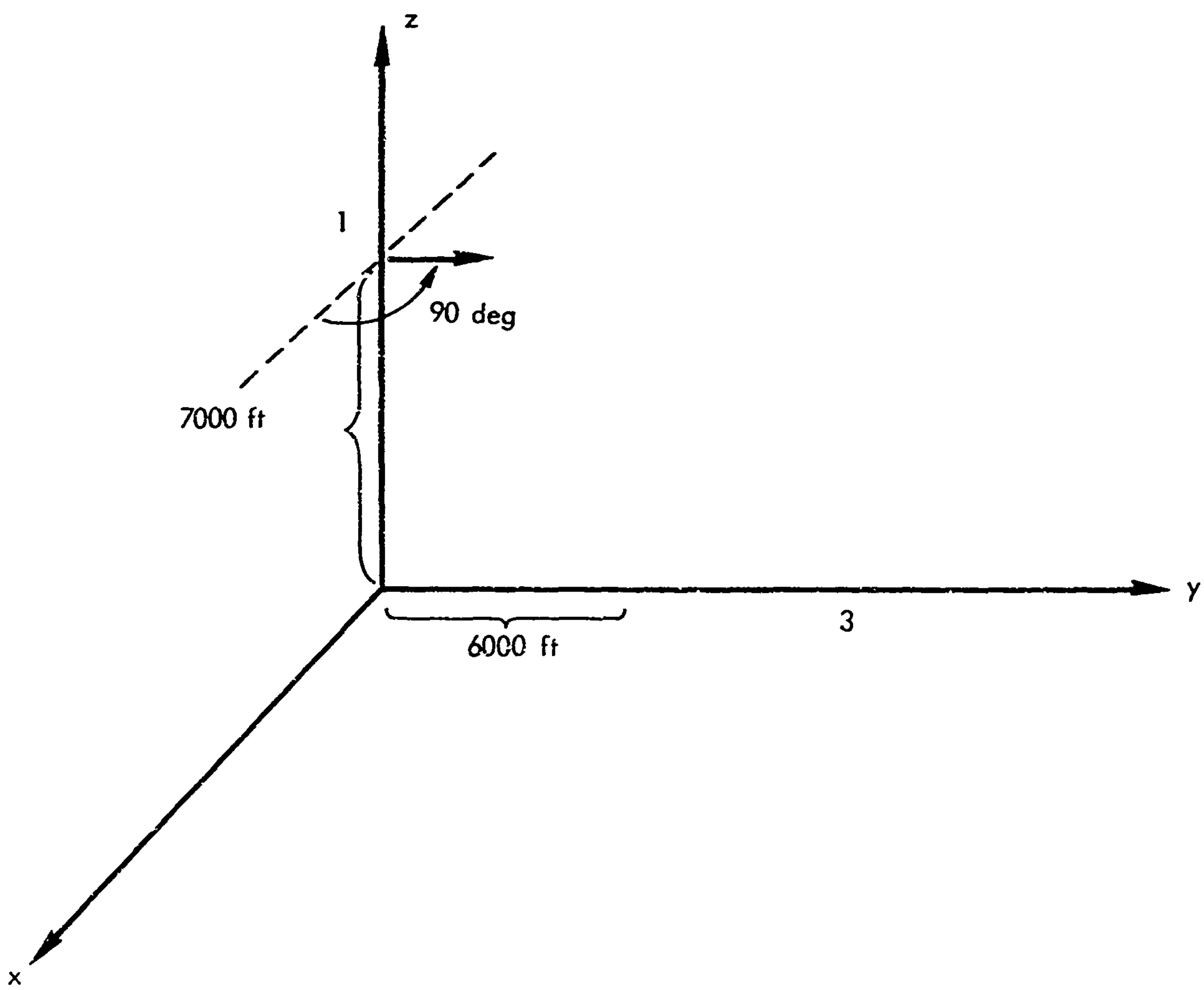


Fig. 28 — Three-dimensional diagram of initial conditions for Example 5

TACTICS PROGRAM - INPUT FORM[a]

VEHICLE TABLE FLAG (0 OR 1)[b]

6	12	18
#1	#2	#3
1	0	0

TITLE CARD[b]

SAMPLE POLICY USING STORE - PROJECTILES

MAIN SET OF INITIAL CONDITIONS

Vehicle #1

1–4	6–18	20–32	34–46	48–60	62–74
0001	Cartesian (0)	x	y	z +.7 E+04	Initial weight +.19470 E+05
0006	Spherical (1)	r	θ	φ	Reference area +.1961 E+03
0011	Cartesian (0)	ẋ	ẏ	ż	a_{Smax} (lateral g's) +.73 E+01
0016	Spherical (1 or 2) +.1 E+01	V +.1 E+04	θ_V +.9 E+02	γ	Mach no. flag

Vehicle #2

1–4	6–18	20–32	34–46	48–60	62–74
0025	Initial weight	Reference area	a_{Smax} (lateral g's)	Captive flight (0,1,2,3) +.1 E+01	Maximum range

Vehicle #3

1–4	6–18	20–32	34–46	48–60	62–74
0040	Cartesian (0)	x	y +.1 E+05	z	Initial weight
0045	Spherical (1)	r	θ	φ	Reference area
0050	Cartesian (0)	ẋ	ẏ	ż	a_{Smax} (lateral g's)
0055	Spherical (1 or 2)	V	θ_V	γ	

PRINT

1–4	6–18	20–32	34–46	48–60	62–74
0062	Initial time	Print interval +.5 E+00	Termination time +.20 E+02		

GUIDANCE (3 VEHICLES)

1–4	Number of time constants	τ(1)	τ(2)	τ(3)	λ navigation constant
0065					+.4 E+02
0070					
0075					

AERODYNAMICS AND PROPULSION (3 VEHICLES)

1–4	C_{Lmax}	C_{D0}	dC_D/dC_L^2	$dC_L/d\alpha$	α_0
0080					
0085					
0090					

1–4	Military thrust	Afterburner thrust	Specific impulse	Burnout weight	Boost acceleration
0095					
0100					
0105					

ROUND-EARTH OPTION[c]

1–4	6–18	20–32	34–46	48–60	62–74
0110	Round-earth flag	Altitude (#1)	Longitude (#1)	Latitude (#1)	Velocity flag (#1)
0115	Rotation (1)	Altitude (#3)	Longitude (#3)	Latitude (#3)	Velocity flag (#3)
0120	Origin longitude	Origin latitude			

MISCELLANEOUS OPTIONS AND EXTRA INPUTS[c]

1–4	6–18	20–32	34–46	48–60	62–74
0127	Integration flag	No. of significant digits req.	Mis. calculation R_{min} +.1 E+04	θ aiming error (#1)	θ_V aiming error (#2)
0127	θ_V aiming error (#3)	γ aiming error (#1)	γ aiming error (#2) +.34 E+01	γ aiming error (#3)	Ballistic coefficient (#1)
0132	Ballistic coefficient (#2)	Ballistic coefficient (#3)			+.1 E-01
d					

[a]The following units are used: distance (ft), time (sec), velocity (ft/sec or Mach no.), acceleration (g's), angles (deg), weight (lb), area (ft^2).

[b]This card required.

[c]Leave this section blank if option not used.

[d]Last data card must have a minus sign in column 1.

Fig. 29 — Input data for Example 5

Aiming error in launching projectiles (DVPHI) = 3.4.

Time interval for printing output (DTPØ) = 0.5.

Time value at which program is to stop (TØTAL) = 20.0.

Missile range to target within which program will automatically initiate process for ground impact miss computation (MINMR) = 1000.0.

Because ballistic constants and variables are defined within the B20MM subroutine, values do not have to be read in. The initial position and velocity of the projectiles are set equal to the fighter values.

Figure 29 shows the initial-condition data for this example as entered on the input form. Refer to Section X for instructions on preparing such a data package.

XIII. NUMERICAL INTEGRATION

MODES OF OPERATION

TACTICS uses an integration subroutine that is an updated and modified version of Adams-Moulton, Runge-Kutta SHARE subroutine RW INT.[(2)] In addition to certain other advantages, this updated version is written in FORTRAN IV source language, whereas RW INT is written in a machine-oriented language. An excellent simplified description of the Adams-Moulton predictor-corrector method of integration is given in Ref. 1, and more detailed information is given in Refs. 2 and 3.

The integration subroutine has four optional modes of operation:

- o Adams-Moulton predictor-corrector method using a variable step size.
- o Runge-Kutta classical fourth-order method using a fixed step size.
- o Adams-Moulton predictor-corrector method using a fixed step size.
- o Adams-Moulton predictor-corrector method using a variable step size, controlled to allow printout exactly at specified intervals.

The optional modes may be selected by setting the flag JINTEG equal to 0, 1, 2, or 3, respectively, either in initial-condition data (DATA 122)* or in PØLICY. If no value is set, JINTEG will automatically be zero and the variable-step-size mode will be used by the program.

Each process has relative advantages. The Adams-Moulton method (JINTEG = 0) generally requires the least execution time, since the step size is automatically controlled (doubled or halved) to maintain a value consistent with a preset truncation error test limit. A possible disadvantage is that substantial overshoots may occur, so that the printout time only approximates the specified time (e.g., 1.22, 2.34, 3.56 sec, instead of 1.0, 2.0, 3.0 sec, respectively). This disadvantage has been removed with only a slight cost in execution

---

*See Appendix E.

time by modifying the variable-step mode of operation so that printout will occur as specified by the user. The fixed-step Adams-Moulton and Runge-Kutta methods generally require the most execution time, but there are certain computational advantages in having a predictable step size. Experience with trial problems indicates that the advantages of both variable- and fixed-step methods may be exploited by changing from one to the other during different phases of the flight simulation, particularly just before and after missile launch. This subject will be discussed in further detail in the following subsections.

INTEGRATION ACCURACY

Double precision procedures are used to reduce round-off errors in accumulating variables. Truncation errors are a function of the step size used. A significant feature of the variable-step Adams-Moulton mode of operation is that the step size is controlled by the integration subroutine so that it will be less than a preset truncation error test limit. This limit, designated ERTEST, is the maximum allowable relative error in any one of the dependent variables in a local region, as distinguished from errors accumulating with each step. Practical values for ERTEST range from 10^{-3} to 10^{-8}; this value is automatically set to 10^{-5} unless otherwise specified by input data or by PØLICY. With reference to the input form of Fig. 9, the value for ERTEST is set by specifying the number of significant digits of accuracy required in data location 123 (for example, if 7 is entered, ERTEST will be set at 10^{-7}). It should be emphasized that the truncation error test limit is only applicable when either of the two variable-step modes of operation is used (JINTEG = 0 or JINTEG = 3).

FIXED-STEP APPROACH TO LAUNCH

As previously indicated, the use of the variable-step options will generally reduce execution time, which will result in an appropriate step size for a specified truncation error. Accordingly, these modes should be used whenever the problem permits--e.g., in time-consuming aerial acrobatics. However, a disadvantage exists due to the inability

to predict which step will be used at a particular phase of the problem, particularly when missile-launch criteria are about to be satisfied. A problem arises because the step may be large, allowing an undesirable overshoot of the criteria (similar to the inexactitude of printout in the JINTEG = 0 mode of operation). A remedy has been provided which automatically causes launch criteria (time, range to target, etc.) to be approached with a fixed minimum step size. Actually, the procedure is to approach launch on a variable (or fixed) step and, when overshoot occurs, back up to the previous time step, restore all conditions, and approach again using small fixed steps. Moreover, these fixed integration steps will be maintained during the missile flight time until the point of closest approach to the target or miss distance has been determined. The previous integration mode of operation (whether fixed or variable) will then be restored unless the option is used to terminate the problem run. If the mode of operation was fixed-step in the first place, back-up and fixed-step approaches to launch would not occur.

MISS-DISTANCE COMPUTATION

The point of closest approach is taken to be the value of the relative missile range to target when its time derivative equals zero. That is, when approaching the target the range rate will be negative, when closest to the target it will be zero, and after passing the target it will be positive. This criterion is used to determine miss distance by "backing up" the integration process in a manner similar to that described for fixed-step approach to launch. The procedure is to return to the previous time step whenever the missile-to-target range rate becomes positive, restore all conditions, shrink the time step by a factor of ten, and repeat this process until the time step becomes smaller than DTMIN (DATA 134). If this value is not read in by the user, it is automatically set to 10^{-5}. In order to allow positive range rates to occur without initiating the process for miss-distance computation, an additional condition is imposed: The missile range to target must be less than MINMR (DATA 124). Otherwise, miss distance will not be calculated. For ASM or SSM applications it is usually necessary

to determine the ground range to target at ground impact (i.e., zero missile altitude), which (as mentioned earlier) is generally not the same as the point of closest approach. TACTICS will automatically compute this ground range by using the altitude of vehicle 2 rather than range rate as the criterion for backing up to determine the time and range at ground impact. This feature is triggered if the altitude of vehicle 3 is exactly 0.0.

INITIALIZATION

The Adams-Moulton predictor-corrector modes of integration, both fixed- and variable-step, use previous values of the variables to predict and to correct values for each new step. In other words, past history is used to predict the future. This procedure is not appropriate in many cases because of discontinuities or large step changes (e.g., a large instantaneous reversal in acceleration from right to left turn, or from climb to dive). Large step changes can not only cause a significant waste of execution time (particularly when the variable-step modes are used) but can also cause important errors. This possibility may be eliminated by reinitializing the integration process whenever interfaces occur, e.g., changes in PØLICY or propulsion. Initializing may be thought of as simply stopping and starting over again, not using previous values for prediction of new values. An automatic reinitializing feature has been built into the framework of the PØLICY subroutine using the integer variables JPØL, KPØL, LPØL, MPØL, and NPØL. A change in any one of these will trigger this process. Accordingly, in formulating a PØLICY subroutine, the user should be careful to separate drastic changes in acceleration, e.g., thrust, boost, or maneuver changes, by using these FORTRAN variables. (See the examples in Section XII.)

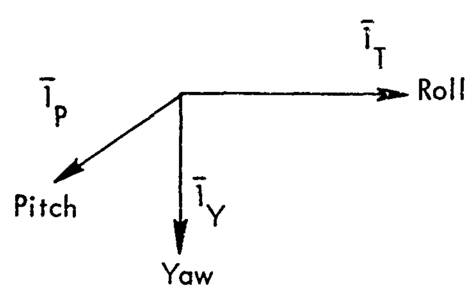
BLANK PAGE

Appendix A

AIRCRAFT ATTITUDE AND ORIENTATION ANGLES

The relationships given in this appendix, contained in subroutine ATITUD, are applicable to conventional airframes where coordinated turns are assumed, as defined in Section IV (see Fig. 7). Thus the use of this version of ATITUD is optional, depending on a particular vehicle's airframe characteristics. For example, if a cruciform missile airframe is to be represented, the attitude-angle relationships corresponding to those in this appendix may be included in the missile control-law routine without calling upon ATITUD, or a different version of the routine may be formulated.

A right-hand set of orthogonal roll, pitch, and yaw orientation axes is assumed as shown below,

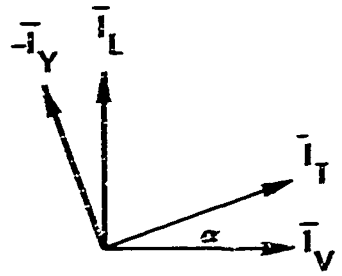


where $\overline{1}_P$ is oriented in the direction of the right wing, $\overline{1}_T$ is oriented in the direction of the thrust vector, and $\overline{1}_Y$ forms a right-hand orthogonal system with $\overline{1}_P$ and $\overline{1}_T$ (approximately in the opposite direction of the lift vector $\overline{L}$, i.e., neglecting the angle of attack α).

YAW AXIS

With the assumption that the vehicle velocity vector is in the plane formed by $\overline{1}_T$, $\overline{1}_Y$ vectors (i.e., no sideslip angle), the unit vector $\overline{1}_Y$ may be determined by examination of the sketch on the following page:

PRECEDING PAGE BLANK



$$\bar{1}_Y = \bar{1}_V \sin \alpha - \bar{1}_L \cos \alpha$$

where, from Fig. 8,

$$\bar{1}_L = \bar{L}/L = \frac{W}{g} \; \frac{a_{oA} \, \bar{1}_A + (a_{oV} + g \cos \gamma) \, \bar{1}_D}{F_a} \tag{45}$$

ROLL AXIS

Similarly, the roll axis $\bar{1}_T$ is defined by

$$\bar{1}_T = \bar{1}_V \cos \alpha + \bar{1}_L \sin \alpha \tag{46}$$

PITCH AXIS

With $\bar{1}_Y$ and $\bar{1}_T$ determined, the unit vector $\bar{1}_P$ may be readily found from the right-hand orthogonal relationship

$$\bar{1}_P = \bar{1}_Y \times \bar{1}_T \tag{47}$$

AZIMUTH AND ELEVATION ANGLES

The unit vectors $\bar{1}_P$, $\bar{1}_Y$, and $\bar{1}_T$ define the attitude or orientation of a vehicle. Now we may conveniently perform a coordinate transformation to resolve the relative range vector $\bar{r}_{ij}$ $(i = 1,2,3;\ j = 1,2,3)$ into this frame of reference. The primary objective is to determine

the orientation of $\bar{r}(i,j)$, the LOS, in terms of an azimuth angle η and an elevation angle $\mathcal{E}$, defined as shown in Fig. 30.

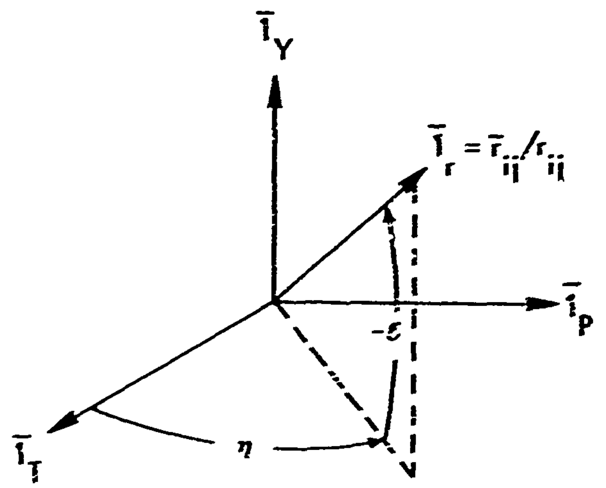


Fig. 30—Azimuth and elevation coordination transformation

The negative sign for $\mathcal{E}$ is an unfortunate result of the conventional aerodynamic orientation of the pitch, roll, and yaw vectors. (We wish $+\mathcal{E}$ to correspond to the up direction.)

The relative range vector $\bar{r}_{ij}$ may be transformed into this system by the following equation:

$$\bar{r}_{ij} = (\bar{r}_{ij} \cdot \bar{1}_T)\bar{1}_T + (\bar{r}_{ij} \cdot \bar{1}_P)\bar{1}_P + (\bar{r}_{ij} \cdot \bar{1}_Y)\bar{1}_Y \qquad (48)$$

where

$$\begin{aligned} \bar{r}_{ij} \cdot \bar{1}_T &= r_{ij} \cos \mathcal{E} \cos \eta \\ \bar{r}_{ij} \cdot \bar{1}_P &= r_{ij} \cos \mathcal{E} \sin \eta \qquad (49) \\ \bar{r}_{ij} \cdot \bar{1}_Y &= r_{ij} \sin \mathcal{E} \end{aligned}$$

From the above, we may conclude that

$$\eta = \tan^{-1} \frac{\bar{1}_r \cdot \bar{1}_P}{\bar{1}_r \cdot \bar{1}_T}$$

$$\mathcal{E} = -\sin^{-1}\left(\bar{1}_r \cdot \bar{1}_Y\right) \tag{50}$$

where

$$\bar{1}_r = \frac{\bar{r}_{ij}}{r_{ij}}$$

BANK ANGLE

The bank angle ϕ_B is defined with respect to a wind axis or velocity-vector reference system (see Figs. 7 and 8 and Eq. (19)). TACTICS calculates the angle from the expression

$$\phi_B = -\tan^{-1} \frac{\bar{1}_L \cdot \bar{1}_D}{\bar{1}_L \cdot \bar{1}_A} \tag{51}$$

where $\bar{1}_L$ is the unit vector along the lift vector $\bar{L}$ (normal to $\bar{V}$).

ROLL ANGLE

Roll angle ϕ (ROLL(I)) is arbitrarily defined as the angle existing between the pitch axis, i.e., the winge of the vehicle, and a line through the c.g. of the vehicle both normal to the longitudinal (i.e., roll) axis and parallel to the horizontal plane. The determination is accomplished as follows. First, the spherical coordinates θ_T and φ_T of the unit vector $\bar{1}_T$ are computed:

$$\theta_T = \tan^{-1} \frac{1_{Ty}}{1_{Tx}}$$

$$\varphi_T = \sin^{-1} (1_{Tz}) \tag{52}$$

Next, the unit vectors $\bar{1}_{A1}$ and $\bar{1}_{D1}$ analogous to the vectors $\bar{1}_A$ and $\bar{1}_D$ are determined. That is, $\bar{1}_{A1}$ is normal to $\bar{1}_T$ and in the horizontal plane, and $\bar{1}_{D1}$ is normal to $\bar{1}_T$ and in a vertical plane so as to form a right-handed orthogonal system. The components of these vectors are

$$
\begin{aligned}
1_{A1x} &= -\sin\theta_T & \qquad 1_{D1x} &= -\sin\varphi_T\cos\theta_T \\
1_{A1y} &= \cos\theta_T & \qquad 1_{D1y} &= -\sin\varphi_T\sin\theta_T \\
1_{A1z} &= 0 & \qquad 1_{D1z} &= \cos\varphi_T
\end{aligned}
\tag{53}
$$

The roll angle ψ may now be computed from the geometry shown in Fig. 31 as viewed from the tip of unit vector $\bar{1}_T$ (directed out of the page).

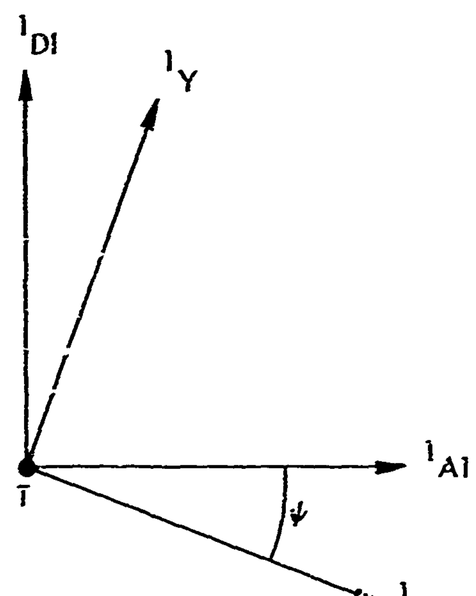


Fig. 31—Roll-angle geometry

$$\psi = \tan^{-1} \frac{\bar{1}_P \cdot \bar{1}_{D1}}{\bar{1}_Y \cdot \bar{1}_{D1}} \tag{54}$$

As defined above, a turn to the right results in a positive roll angle varying from 0 to 180 deg, and a turn to the left results in a negative roll angle varying from 0 to 180 deg.

Appendix B

INTEGRATING THE EQUATIONS OF MOTION

In representing the flight of three vehicles in motion simultaneously, there are 18 differential equations to be numerically integrated, nine involving the accelerations $\dot{\overline{V}}_{(i)}$ and the other nine involving the velocities $\overline{V}_{(i)}$. These equations might be expressed in a variety of different forms, each having certain advantages in certain situations. The TACTICS program expresses the differential equations in two different forms: One involves the flat-earth representation (less complex and faster), and the other involves a round rotating or nonrotating earth (essential for space applications, long ranges, or high speeds). Unless specified by the KINTEG option, DATA(110), the flat-earth equations, are used. In order to decrease execution time, only 12 equations are integrated when vehicle 2 is in captive flight (subroutine CAPFLT).

FLAT-EARTH CARTESIAN FORM

The accelerations are determined by resolving the applied forces (e.g., aerodynamic, gravitational, or propulsive) into three components. Two of these components, a_{oh} and a_{oV}, are normal to the velocity vector $\overline{V}$, and their vector sum is the net lateral acceleration of the vehicle. The direction of the third component, V, is parallel to or along the velocity vector $\overline{V}$. When integration is to occur, the three components of acceleration a_{oh}, a_{oV}, and $\dot{V}$, as well as the unit vectors $\overline{1}_A$, $\overline{1}_D$, and $\overline{1}_V$, have been determined, so that the expression for the net acceleration vector $\overline{a}_o$ is

$$\overline{a}_o = \overline{a}_{oh} \, \overline{1}_A + a_{oV} \, \overline{1}_D + \dot{V} \, \overline{1}_V \tag{55}$$

Converting to Cartesian coordinates yields the expression

$$\ddot{x} = a_{oh} \, 1_{Ax} + a_{oV} \, 1_{Dx} + \dot{V} \, 1_{Vx} \tag{56}$$

and similarly for $\ddot{y}$, $\ddot{z}$. (See Eqs. (9).)

Integration of the equations for $\bar{a}_1$, $\bar{a}_2$, and $\bar{a}_3$ corresponding to each of the three vehicles will result in $\bar{V}_1$, $\bar{V}_2$, and $\bar{V}_3$, respectively. At this point, the use of *relative* velocities and positions is introduced to minimize errors in relative position due to differencing large numbers. Accordingly, the velocities of vehicles 1 and 2 relative to vehicle 3 are formed and integrated to obtain the relative positions. The equations are as follows:

$$\begin{aligned} \bar{V}_{13} &= \bar{V}_3 - \bar{V}_1 \\ \bar{V}_{23} &= \bar{V}_3 - \bar{V}_2 \end{aligned} \tag{57}$$

Integration of the above equations yields the relative position vectors $\bar{r}_{13}$ and $\bar{r}_{23}$. The velocity of vehicle 3, $\bar{V}_3$ (i.e., an absolute velocity relative to the fixed origin of the x, y, z frame), is integrated to yield the absolute position vector $\bar{r}_3$. The absolute positions $\bar{r}_1$ and $\bar{r}_3$ are subsequently determined from

$$\begin{aligned} \bar{r}_1 &= \bar{r}_3 - \bar{r}_{13} \\ \bar{r}_2 &= \bar{r}_3 - \bar{r}_{23} \end{aligned} \tag{58}$$

This procedure emphasizes the accuracy of the relative rather than absolute position geometry.

ROUND-EARTH SPHERICAL FORM

The dynamic equations for a particle whose motion is observed from a rotating reference frame involve centrifugal and Coriolis-force terms. Moreover, since the earth is represented as spherical, a number of coordinate transformations are necessary to relate the various frames of reference. For convenience, the basic equations of motion and certain unit vector notations have been extracted from the ROCKET program[(1)] in order to provide compatibility. However, since TACTICS is primarily concerned with the relative motion of possibly three

different vehicles, there are significant differences between coordinate systems and computational methods employed.

The basic differential equations of motion are derived using three unit vectors defined as follows:

$\overline{1}_R$, directed radially from the earth's center to the vehicle, a point mass.

$\overline{1}_L$, normal to $\overline{1}_R$ and directed eastward along a parallel of latitude.

$\overline{1}_P$, directed northward along the local meridian so as to form a right-handed orthogonal system with $\overline{1}_R$ and $\overline{1}_L$.

The acceleration of the vehicle with respect to a nonrotating inertial X, Y, Z coordinate system is

$$\overline{a} = \ddot{\overline{r}} = \ddot{X}\,\overline{i} + \ddot{Y}\,\overline{j} + \ddot{Z}\,\overline{k} \tag{59}$$

This acceleration $\overline{a}$ corresponds to a net result of all forces applied, $\overline{F}/m$. The following derivation develops expressions for the net acceleration resulting from $\overline{F}$ in terms of a rotating-earth fixed-coordinate system, i.e., longitude λ, latitude φ, radial distance r, and time derivatives thereof.

First, the following simple device may be used to transform coordinates to the $\overline{1}_R$, $\overline{1}_L$, $\overline{1}_P$ system:

$$\overline{a} \cdot \overline{1}_R = (\overline{1}_R \cdot \overline{i})\,\ddot{X} + (\overline{1}_R \cdot \overline{j})\,\ddot{Y} + (\overline{1}_R \cdot \overline{k})\,\ddot{Z} \tag{60}$$

and similarly for $\overline{a} \cdot \overline{1}_L$ and $\overline{a} \cdot \overline{1}_P$.

Referring to Fig. 32, the required dot products are by inspection

$$\begin{array}{lll}
\overline{1}_R \cdot \overline{i} = \cos\varphi\cos\theta & \overline{1}_L \cdot \overline{i} = -\sin\theta & \overline{1}_P \cdot \overline{i} = -\sin\varphi\cos\theta \\
\overline{1}_R \cdot \overline{j} = \cos\varphi\sin\theta & \overline{1}_L \cdot \overline{j} = \cos\theta & \overline{1}_P \cdot \overline{j} = -\sin\varphi\sin\theta \\
\overline{1}_R \cdot \overline{k} = \sin\varphi & \overline{1}_L \cdot \overline{k} = 0 & \overline{1}_P \cdot \overline{k} = \cos\varphi
\end{array} \tag{61}$$

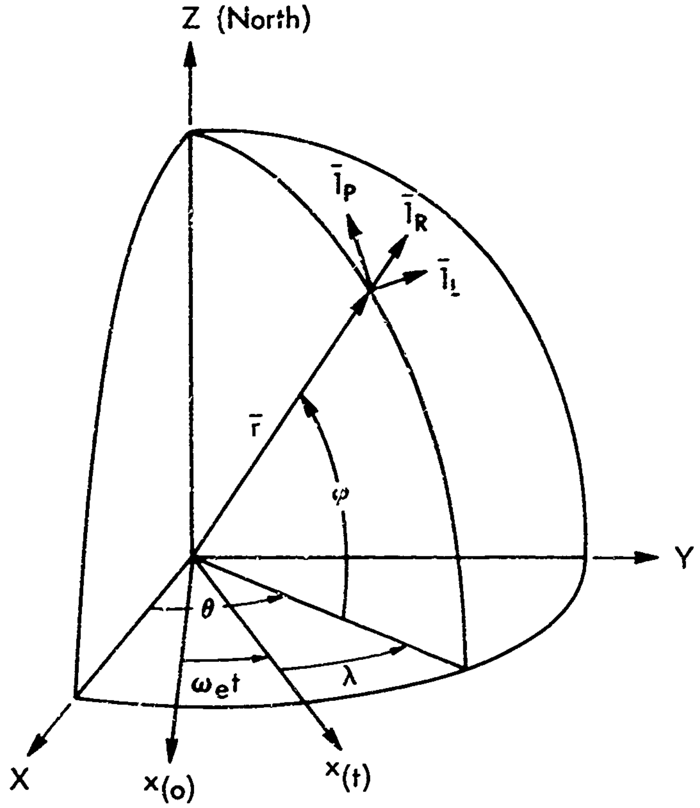


Fig. 32 — Geocentric coordinate system

(Note the analogy with the definition of unit vectors $\overline{1}_V$, $\overline{1}_A$, and $\overline{1}_D$; see Eq. (6).)

The desired transformation may now be accomplished using

$$\overline{a} = (\overline{a} \cdot \overline{1}_R)\ \overline{1}_R + (\overline{a} \cdot \overline{1}_L)\ \overline{1}_L + (\overline{a} \cdot \overline{1}_P)\ \overline{1}_P \tag{62}$$

Expanding and substituting the dot products results in

$$\begin{aligned} \overline{a} = \overline{1}_R\ & (\ddot{X} \cos\varphi \cos\theta + \ddot{Y} \cos\varphi \sin\theta + \ddot{Z} \sin\varphi) \\ & + \overline{1}_L\ (-\ddot{X} \sin\theta + \ddot{Y} \cos\theta + 0) \\ & + \overline{1}_P\ (-\ddot{X} \sin\varphi \cos\theta - \ddot{Y} \sin\varphi \sin\theta + \ddot{Z} \cos\varphi) \end{aligned} \tag{63}$$

The inertial coordinates X, Y, Z may be expressed in terms of spherical coordinates as

$$\begin{aligned} X &= r \cos\varphi \cos\theta \\ Y &= r \cos\varphi \sin\theta \\ Z &= r \sin\varphi \end{aligned} \tag{64}$$

Differentiating these expressions twice with respect to time and substituting in Eq. (63) will yield

$$\begin{aligned} \overline{a} = \overline{1}_R\ & (\ddot{r} - r\,\dot{\varphi}^2 - r\,\dot{\theta}^2 \cos^2\varphi) \\ & + \overline{1}_L\ (\ddot{\theta}\, r \cos\varphi + 2\,\dot{r}\,\dot{\theta} \cos\varphi - 2\,r\,\dot{\varphi}\,\dot{\theta} \sin\varphi) \\ & + \overline{1}_P\ (\ddot{\varphi}\, r + r\,\dot{\theta}^2 \sin\varphi \cos\varphi + 2\,\dot{r}\,\dot{\varphi}) \end{aligned} \tag{65}$$

Referring to Fig. 32, we note that θ is a time variable:

$$\theta = \lambda + \omega_e t$$

where

λ = an arbitrarily defined angle of longitude on the earth's surface

ω_e = the earth's angular rate of rotation ($7.29211585 \cdot 10^{-5}$ rad/sec)

t = the time interval of rotation

Accordingly,

$$\dot{\theta} = \dot{\lambda} + \omega_e \tag{66}$$

and

$$\ddot{\theta} = \ddot{\lambda} \tag{67}$$

The resultant force $\overline{F}$, causing acceleration $\overline{a}$ with respect to the inertial X, Y, Z frame, may be resolved into components F_R, F_L, F_P in the $\overline{1}_R$, $\overline{1}_L$, and $\overline{1}_P$ directions, respectively. Substituting $\overline{a} = \overline{F}/m$ in the preceding equations and equating components results in the desired set of equations of motion:

$$\ddot{r} = F_R/m + r\,\dot{\varphi}^2 + r\,(\dot{\lambda} + \omega)^2 \cos^2 \varphi$$

$$\ddot{\lambda} = \left[F_L/m - 2\,\dot{r}\,(\dot{\lambda} + \omega) \cos \varphi + 2\,r\,\dot{\varphi}\,(\dot{\lambda} + \omega) \sin \varphi\right] / r \cos \varphi$$

$$\ddot{\varphi} = \left[F_P/m - r\,(\dot{\lambda} + \omega)^2 \sin \varphi \cos \varphi - 2\,\dot{r}\,\dot{\varphi}\right] / r \tag{68}$$

These equations are a set of second-order differential equations in terms of geocentric radial distance r, longitude λ, latitude φ, and components of resultant force $\overline{F}$. (The methodology of force definition is discussed in Section IV.) Numerical integration of these second-order equations will yield the first-order velocity components $\dot{r}$, $\dot{\lambda}$, and $\dot{\varphi}$ at a time $t + \Delta t$. Theoretically, a second integration will yield the position quantities r, λ, and φ, but we wish to avoid this procedure to minimize numerical errors arising from differencing large numbers associated with the geocentric distance r. Accordingly, position components are computed by the following method:

- Integrating the second-order equations to determine $\dot{r}$, $\dot{\lambda}$, and $\dot{\varphi}$.
- Forming the topocentric velocities $\dot{\bar{\rho}}_{(i)}$, i.e., the velocities with respect to an arbitrarily chosen reference point (or coordinate system origin) on the earth's surface.
- Integrating the topocentric velocities $\dot{\bar{\rho}}_{(i)}$ to determine the topocentric range position vectors $\bar{\rho}_{(i)}$.

The topocentric reference point or local coordinate system origin $\bar{R}_o$ is defined by input data in terms of latitude φ_o and longitude λ_o so that

$$R_{ox} = R_o \cos \varphi_o \cos (\lambda_o + \omega_e t)$$

$$R_{oy} = R_o \cos \varphi_o \sin (\lambda_o + \omega_e t) \tag{69}$$

$$R_{oz} = R_o \sin \varphi_o$$

where R_o is taken to be 2.0925861×10^7 ft.

The velocity of $\bar{R}_o$ with respect to an inertial frame is

$$\dot{\bar{R}}_o = - R_{oy} \omega_e \bar{i} + R_{ox} \omega_e \bar{j} \tag{70}$$

Note that in the second-order differential equations the centrifugal and Coriolis accelerations are a function of latitude angle φ; hence, numerical solutions will be affected by assumed initial-condition values for vehicle latitude. On the other hand, initial assumptions for vehicle longitude position may be purely arbitrary. Figure 33 shows the topocentric nonrotating coordinate system with the X_T, Y_T, Z_T axis always remaining parallel to the X, Y, Z inertial axis. With the $\bar{R}_o$ origin established, the topocentric range vectors (from the topocentric origin to the vehicles) are

$$\bar{\rho}_i = \bar{r}_i - \bar{R}_o \qquad i = 1, 2, 3 \tag{71}$$

and the velocities are

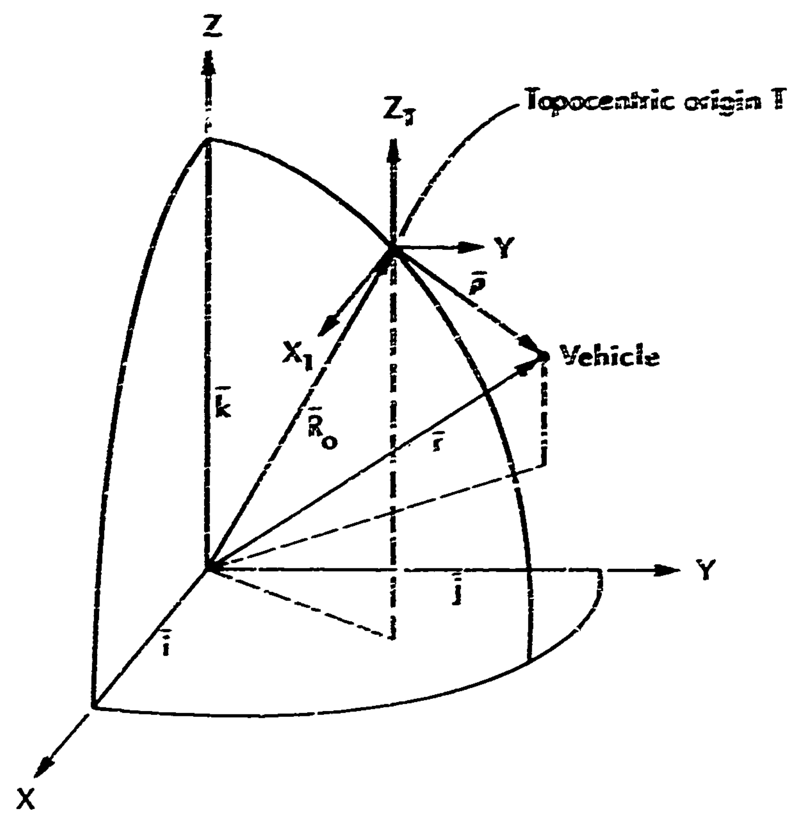


Fig. 33 — Topocentric coordinate system

$$\dot{\bar{\rho}}_i = \dot{\bar{r}}_i - \dot{\bar{R}}_o$$

$$\dot{\bar{r}}_i = \left.\frac{d\bar{r}}{dt}\right|_{local} + \bar{\omega}_e \times r \tag{72}$$

where

$$\left.\frac{d\bar{r}}{dt}\right|_{local} = \dot{r}\,\bar{1}_R + r\,\dot{\lambda}\cos\varphi\,\bar{1}_L + r\,\dot{\varphi}\,\bar{1}_P \tag{73}$$

and

$$\omega_e \times \bar{r} = r\,\omega_e \cos\varphi\,\bar{1}_L \tag{74}$$

so that

$$\dot{\bar{r}}_i = \dot{r}_R + r(\dot{\lambda} + \omega_e)\cos\varphi\,\bar{1}_L + r\,\dot{\varphi}\,\bar{1}_P \tag{75}$$

The integration of the topocentric velocities $\dot{\bar{\rho}}_i$ will yield the positions $\bar{\rho}_i$ in terms of the inertial coordinates X_T, Y_T, Z_T. Although this coordinate system is convenient for astronomical or space applications, it is not desirable for many other problems in which the horizontal plane and the local vertical are more significant. Referring back to the flat-earth x, y, z system, we would like the z-component to correspond to altitude and the z-y plane to correspond approximately to the horizontal plane. Accordingly, all $\bar{\rho}_i$ coordinates are transformed into an azimuth-elevation system with the origin at the $\bar{R}_o$ position, the z-axis in the direction of $\bar{R}_o$, and the y-axis directed eastward, as shown in Fig. 34. Subroutine TØPØCN is used for transforming from either equatorial plane to azimuth-elevation coordinates or vice versa.

The coordinate transformations are as follows. The transformation from topocentric X_T, Y_T, Z_T coordinates to azimuth-elevation x, y, z coordinates is given by the following matrix:

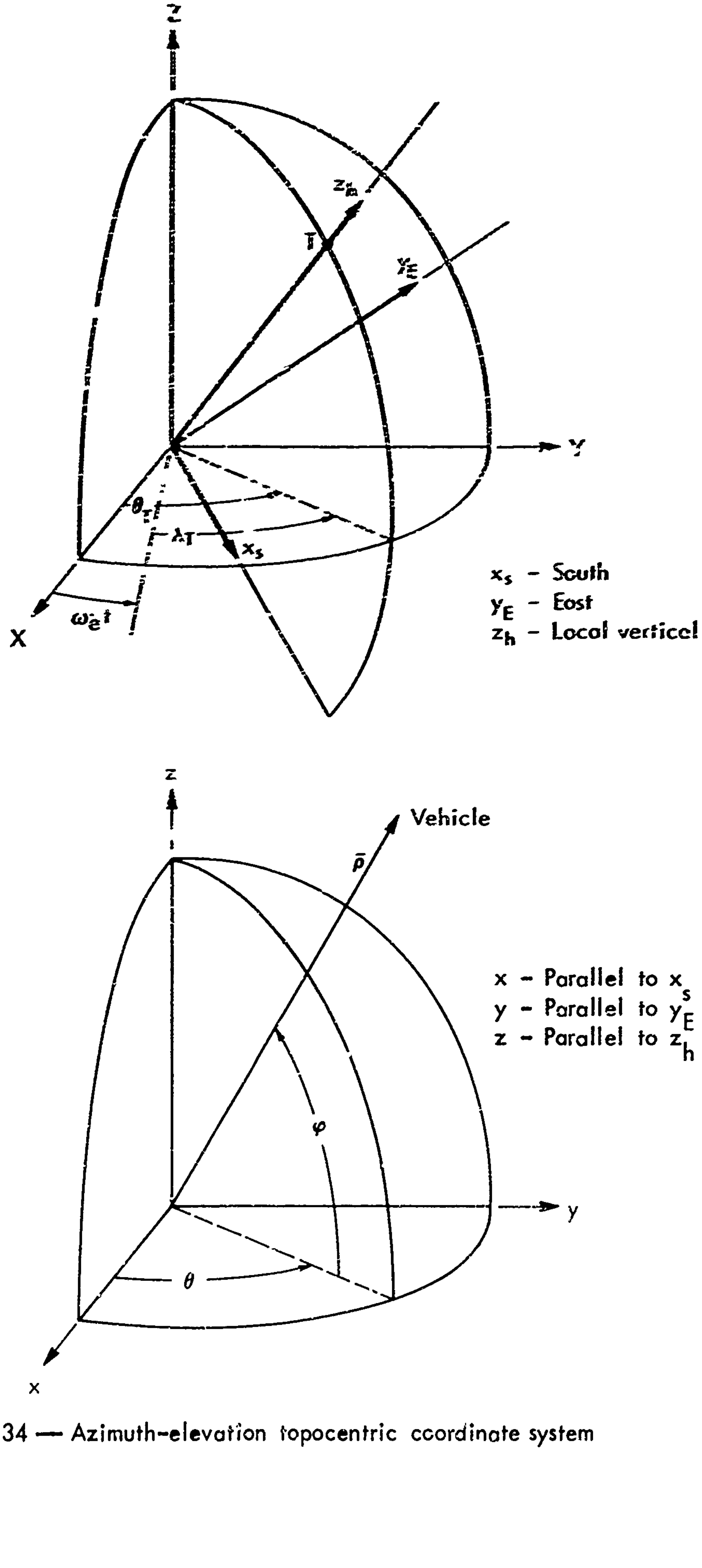


Fig. 34 — Azimuth-elevation topocentric coordinate system

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \sin\varphi_T \cos\theta_T & \sin\varphi_T \sin\theta_T & -\cos\varphi_T \\ -\sin\theta_T & \cos\theta_T & 0 \\ \cos\theta_T \cos\varphi_T & \cos\varphi_T \sin\theta_T & \sin\varphi_T \end{bmatrix} \begin{bmatrix} X_T \\ Y_T \\ Z_T \end{bmatrix} \quad (76)$$

The reverse transformation, from azimuth-elevation x, y, z coordinates to topocentric X_T, Y_T, Z_T coordinates, is given by

$$\begin{bmatrix} X_T \\ Y_T \\ Z_T \end{bmatrix} = \begin{bmatrix} \sin\varphi_T \cos\theta_T & -\sin\theta_T & \cos\theta_T \cos\varphi_T \\ \sin\varphi_T \sin\theta_T & \cos\theta_T & \sin\theta_T \cos\varphi_T \\ -\cos\varphi_T & 0 & \sin\varphi_T \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (77)$$

where φ_T and θ_T are the latitude and longitude, respectively, of the rotating reference origin at point T, defined by

$$\begin{aligned} \varphi_T &= \varphi_T(t_o) \\ \theta_T &= \lambda_T(t_o) + \omega_e\,(t - t_o) \end{aligned} \quad (78)$$

In addition to the "standard" printout in terms of x, y, z azimuth-elevation (horizontal-plane) system coordinates, the quantities of latitude φ, longitude λ, and altitude h for each vehicle are available as an optional printout section. Note that because of the earth curvature and the use of a fixed-earth reference point ($\bar{R}_o$), the magnitude of the z-position component will not generally correspond exactly to altitude (because of the x-y displacement from the origin).

Appendix C

GUIDANCE-AND-CONTROL-LAW DEFINITIONS

This appendix contains descriptions and mathematical definitions of most of the significant guidance and control laws developed for use with TACTICS. The list is open-ended and incomplete, since new ideas are continuously being implemented as new subroutines. See Section V for a summary of definitions and the applicable acceleration equations.

OPEN-LOOP CONTROL LAWS

Straight Flight

The commanded lateral acceleration $\overline{a}_c$ is zero. The vehicle will fly a straight-line path (but not necessarily "straight and *level*"). However, an acceleration or deceleration *along* this path may occur due to the thrust-drag relationship.

Straight and Level Flight

This routine is used to return a vehicle to level flight in the horizontal plane from a diving or climbing condition. An arbitrary control-law assumption is made that the vehicle will commence the return to level flight at maximum g capability subject to structural or C_{Lmax} constraints. The magnitude of this acceleration tapers off linearly as the velocity vector reaches an angle of 10 deg from the horizontal plane and becomes zero within 0.1 deg of level flight. The direction of the acceleration is accordingly

$$\overline{1}_1 = \pm \overline{1}_D \tag{79}$$

(positive for climbing and negative for diving). The magnitude of the acceleration follows the arbitrary law

$$a_C = \min\left(a_{S\max};\ \left|(\gamma/10)a_{S\max}\right|\right)$$
$$a_C = 0 \qquad \gamma < 0.1 \tag{80}$$

Captive Flight

This routine is used to zero out computations and printed values for vehicles which are in captive flight. There are three modes:

- Vehicle 2 locked to vehicle 1.
- Vehicle 1, 2, or 3 locked to the zero origin (i.e., all values for position, velocity, and acceleration are zero).
- Vehicle 2 locked to vehicle 3.

Captive flight is not a guidance law in the sense of the preceding discussion but rather a device to eliminate unnecessary computations and improve the appearance of the printed values.

Launch

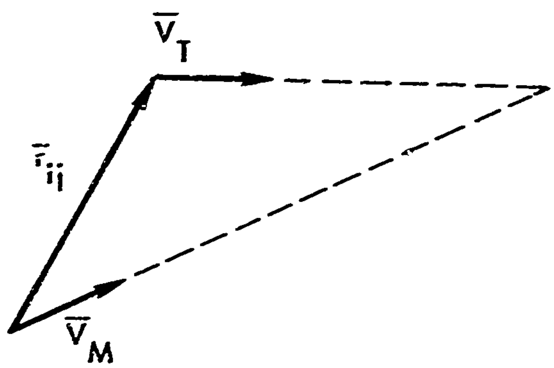
This control law, which simulates the launch-boost phase of a missile flight, may be applied to any of the three vehicles. The call for "launch" is usually based on some criteria stated in PØLICY (e.g., range, range rate, geometry, accelerations, time, and--most importantly--combinations thereof). When this routine is called, the boost velocity ΔV must be specified as a constant. The commanded lateral acceleration is gravitational only:

$$\overline{a}_C = -g \cos \gamma\ \overline{1}_D \tag{81}$$

Aim (1)

This routine was provided for applications in which it is necessary to simulate the aiming process, e.g., SAM launch. It involves the following problem: Given a target velocity vector $\overline{V}_T$ and a vehicle speed V_M (both assumed to remain constant) separated by a range vector $\overline{r}_{ij}$, find the orientation of $\overline{V}_M$, i.e., aiming angles, so that an

intercept will occur. Like captive flight, it is a device for defining angles rather than a law for defining $\bar{a}_C$. The solution is outlined below:



$$\bar{r}_{ij} + \bar{V}_T T = \bar{V}_M T \qquad (82)$$

where T is the "time-to-go" until impact. The magnitude of $\bar{V}_M$ is known, but the components V_{Mx}, V_{My}, and V_{Mz} are to be determined. From the above equation,

$$|\bar{r}_{ij} + \bar{V}_T T| = \left(r_{ij}^2 + 2\bar{r}_{ij} \cdot \bar{V}_T T + V_T^2 T^2\right)^{\frac{1}{2}}$$

$$T = \frac{\left(-b \pm \sqrt{b^2 - 4ac}\right)}{2a} \qquad T > 0 \qquad (83)$$

where a, b, and c are the corresponding coefficients in the quadratic equation. Hence,

$$\bar{V}_M = \frac{\bar{r}_{ij}}{T} + \bar{V}_T$$

$$V_{Mx} = \frac{r_{ij_x}}{T} + V_{Tx} \qquad (84)$$

and similarly for V_{My} and V_{Mz}.

The aiming angles φ_M and θ_M may be determined from

$$\theta_M = \tan^{-1}\left(\frac{V_{My}}{V_{Mz}}\right)$$
$$\varphi_M = \sin^{-1}\left(V_{Mz}\right) \tag{85}$$

<u>Aim (2)</u>

This routine is similar in purpose to the previous item. In certain problems it is necessary to simulate the launching of a missile (or firing of a projectile) along the LOS to the target. The aiming process is more complicated if the launching platform motion is taken into account. In order for the projectile to travel along the LOS (ignoring ballistic drop), the orientation angles of the *resultant* projectile velocity and platform velocity must be the same as those of the LOS. Hence, if $\overline{V}_2$ is the projectile velocity and $\overline{V}_1$ is the launching platform velocity, $\overline{V}_{12} = \overline{V}_2 + \overline{V}_1$, where $\overline{V}_{12}$ is the resultant velocity from which

$$\overline{V}_2 = \overline{V}_{12} - \overline{V}_1$$
$$V_2^2 = V_{12}^2 - 2\overline{V}_1 \cdot \overline{V}_{12} - V_1^2 \tag{86}$$

The magnitude of V_{12} may now be determined from

$$V_{12}^2 - 2V_1V_{12}\cos\left(\theta_{LOS} - \theta_{V1}\right)\cos\left(\varphi_{LOS} - \gamma_{V1}\right) - V_1^2 - V_2^2 = 0 \tag{87}$$

Knowing the magnitude and orientation of $\overline{V}_{12}$,

$$V_{2x} = V_{12x} - V_{1x}$$
$$V_{2y} = V_{12y} - V_{1y} \tag{88}$$
$$V_{2z} = V_{12z} - V_{1z}$$

The orientation angles of $\overline{V}_2$ are

$$\gamma_{V2} = \frac{\left(V_{12} \sin \varphi_{LOS} - V_{1x}\right)}{V_2}$$

$$\theta_{V2} = \frac{\left(V_{12} \cos \varphi_{LOS} \sin \theta_{LOS} - V_{2y}\right)}{\left(V_{12} \cos \varphi_{LOS} \cos \theta_{LOS} - V_{2x}\right)} \tag{89}$$

and the magnitude is the boost velocity

$$|\overline{V}_2| = \Delta V$$

The incremental angles which should be added to the launching platform angles for aiming are then

$$\Delta\gamma_{V2} = \gamma_{V2} - \gamma_{V1}$$

$$\Delta\theta_{V2} = \theta_{V2} - \theta_{V1} \tag{90}$$

Left or Right Turn (1)

The commanded lateral acceleration vector $\overline{a}_C$ is in the horizontal plane and has a constant value as specified when calling the routine(s). The two routines (left and right) are identical except for an algebraic sign corresponding to the direction of the turn ($\pm \overline{I}_A$). It was decided to specify the magnitude of the turning acceleration in terms of the resultant normal acceleration, expressed in number of g's or F_n/W (see Eq. (21)). Accordingly,

$$\overline{a}_C = \left(\frac{g}{W}\right)\sqrt{F_n^2 - (W \cos \gamma)^2} \qquad (\pm \overline{I}_A) \tag{91}$$

Left or Right Turn (2)

The commanded lateral acceleration vector $\overline{a}_C$ is in the horizontal plane. The magnitude of $\overline{a}_C$ depends on the excess thrust available to perform a turn with constant Mach number. In other words, these

maneuvers are constant-Mach-number horizontal turns; a necessary aerodynamic condition for initiating such a turn is that thrust must exceed drag, because additional drag will be induced by the maneuver itself. These routines require iterative processes if aerodynamic tables are involved, since the tables must be read in reverse (see item 2 below). A brief outline of the problem is as follows:

1. Since Mach number is constant with altitude, $\dot{M} = \dot{V} = 0$, and the necessary aerodynamic relationship is

$$D + W \sin \gamma + T \cos \alpha = 0 \tag{92}$$

from which D may be calculated. Knowing D,

$$C_D = \frac{D}{Aq} \tag{93}$$

2. Find C_L and α corresponding to C_D and Mach number. For example, using the familiar analytic expressions

$$C_D = C_{D_o} + \frac{dC_D}{d(C_L^2)} C_L^2$$

$$C_L = \sqrt{(C_D - C_{D_o}) \Big/ \frac{dC_D}{d(C_L^2)}}$$

$$\alpha = \frac{C_L}{(dC_L/d\alpha)} + \alpha_o \tag{94}$$

$$L = C_L Aq$$

$$F_n = L + T \sin \alpha$$

The equivalent of the preceding operations must be performed if tabular values are used. In effect, values for $dC_L/d\alpha$ and $dC_D/d(C_L^2)$ must be numerically determined.

3. Find the magnitude and direction of $\bar{a}_C$ corresponding to the available force $\bar{F}_n$.

$$\bar{a}_C = \left(\frac{g}{W}\right)\sqrt{F_n^2 - (W\cos\gamma)^2} \qquad (\pm\,\bar{I}_A) \tag{95}$$

where (+) is used for a left turn and (-) for a right turn. Note that the magnitude of $\bar{a}_C$ is not explicitly specified but is dependent on F_n and γ, i.e., it is not known a priori.

Left or Right Turn (3)

These maneuvers are similar to the Left or Right Turn (2) maneuvers described above except for the constant-altitude requirement. By means of a dive, gravity may be utilized to hold Mach number constant during the turn. If there is insufficient thrust to meet the conditions for constant Mach number and a specified number of g's in the turn, the vehicle will dive in order to satisfy these conditions. On the other hand, if excess thrust is available (or after a sustained dive), the vehicle will climb. If a maximum lift coefficient (C_{Lmax}) limit is reached, the vehicle will turn at the corresponding allowable acceleration for C_{Lmax}, and Mach number will equal a constant. Since Mach number varies with altitude, the following relationship must be satisfied:

$$\dot{M} = \dot{V}/s - \frac{V\dot{s}}{s^2} = 0 \tag{96}$$

Referring to the relationships given in Appendix I and performing the necessary arithmetical operations,

$$\dot{p} = -\,6.9697419 \times 10^{-8}(1 - 6.8865741)^{3.256}\,\dot{z}$$

$$\dot{\rho} = -\,32.207899\,\rho\dot{z} \tag{97}$$

$$\dot{s} = 0.5929\left(\frac{\dot{p}}{\sqrt{\rho p}} - \frac{\sqrt{p\dot{\rho}}}{\rho^{3/2}}\right)$$

From the preceding expression for $\dot{M} = 0$,

$$\dot{V} - V\dot{s}/s = 0 \tag{98}$$

The necessary aerodynamic relationship is

$$D + W \sin \gamma + T \cos \alpha - \frac{V\dot{s}}{s}\left(\frac{W}{g}\right) = 0 \tag{99}$$

from which the drag D may be determined by means of the applicable expressions as given by Eq. (26). The components of commanded acceleration are

$$\begin{aligned} a_{Ch} &= \pm \left(\frac{g}{W}\right) \min\ (F_n,\ F'_n) \\ a_{CV} &= \sqrt{(g/W)\ F_n^2 - a_{ch}^2 - g \cos \gamma} \end{aligned} \tag{100}$$

where F'_n (or rather F'_n/W in terms of number of g's) is to be specified when the maneuver is called and F_n corresponds to the L, C_D, C_L, and α relationships of Eq. (27). The component a_{Ch} is positive for left turns and negative for right turns. Special cases arise if the specified acceleration F'_n/W exceeds structural or aerodynamic constraints. If F'_n should exceed the structural constraints, the value is altered to correspond to the maximum value a_{Smax}. If F'_n exceeds the aerodynamic value

$$F_{nmax} = C_{Lmax}\ Aq + T \sin \alpha_{max} \tag{101}$$

then F'_n is altered to correspond to F_{nmax}. Since C_{Lmax} varies with Mach number and by definition the Mach number should not change, F'_n will remain equal to F_{nmax} under this constraint.

<u>Left or Right Turn (4)</u>

These maneuvers are identical to the Left or Right Turn (3) maneuvers except under the F_{nmax} constraint discussed above. Since C_{Lmax}

varies with Mach number, a correspondence must be established between the specified Mach number and the specified number of g's in the turn. The Left or Right Turn (3) maneuvers limit the g's to the corresponding Mach number and C_{Lmax} condition. The Left or Right Turn (4) maneuvers allow the vehicle to dive and increase Mach number (and hence C_{Lmax}) in order to obtain the specified number of g's in the turn. That is, if the specified F_n' exceeds F_{nmax}, the constant-Mach-number maneuver requirement is initially abandoned to increase speed and F_{nmax} so that

$$F_{nmax} \geq F_n'$$

Under the constraint conditions, the initial dive is defined arbitrarily by

$$\begin{aligned} a_{Ch} &= 0 \\ A_{CV} &= -\ 2.5\ g \cos \gamma \end{aligned} \tag{102}$$

Left or Right Turn (5)

These are climbing or diving turns as specified by arguments of the number of g's required, i.e., F_n/W and the bank angle ψ_B. As shown in Figs. 7 and 8, the commanded acceleration components are

$$\begin{aligned} a_{Ch} &= -\ g\left(\frac{F_n}{W}\right) \sin \psi_B \\ a_{CV} &= g\left(\frac{F_n}{W}\right) \cos \psi_B - g \cos \gamma \end{aligned} \tag{103}$$

A right turn results for positive values from 0 to 180 deg and a left turn results for negative ψ_B values from 0 to -180 deg. Diving or climbing occurs for absolute-value magnitudes greater or less than 90 deg, respectively.

Climb or Dive (1)

These maneuvers are identical to the Left or Right Turn (1) described above except that the acceleration vector $\overline{a}_C$ is in the vertical plane. That is,

$$\overline{a}_C = \pm\, g\left(\frac{F_n}{W} + \cos\gamma\right)\overline{1}_D \qquad (104)$$

The plus sign corresponds to the climb maneuver and the minus sign to the dive maneuver.

Climb (2)

This maneuver is a constant-Mach-number climb depending on excess thrust available at the time it is initiated. The procedure is the same as that for Left or Right Turn (2) described above except that the vector $\overline{a}_C$ is in the vertical plane. That is,

$$\overline{a}_C = g\left(\frac{F_n}{W} - \cos\gamma\right)\overline{1}_D \qquad (105)$$

where F_n is not specified but rather calculated from the relationships described in Left or Right Turn (2) and (3) (see Eq. (94)).

Barrel Roll (1)

In attempting to describe the trajectory of a barrel roll within the framework of coordinated turn definition (as described in Section IV), there does not seem to be universal agreement among pilots and engineers as to precise mathematical definition. The following equations were extracted from a Target Generator Program received from Eglin Air Force Base:*

$$\begin{aligned} a_{Ch} &= -\,g\left(\frac{F_n}{W}\right)\sin\psi_B \\ a_{CV} &= g\left(\frac{F_n}{W}\right)\cos\psi_B - g\cos\gamma \end{aligned} \qquad (106)$$

*Private communication.

where the bank angle ψ_B is obtained from a specified banking rate $\dot{\psi}_B$ so that

$$\psi_B = \int \dot{\psi}_B \, dt \tag{107}$$

Referring to Fig. 8, we note that the force F_n rotates about the velocity vector $\overline{V}$. The acceleration component a_{Ch} varies sinusoidally, whereas the component a_{CV} has a varying gravitational term g cos γ. In calling for Barrel Roll (1), it is necessary to specify (1) the number of 360-deg rolls required, (2) the number of g's for F_n/W, and (3) the banking rate $\dot{\psi}_B$ in degrees per second.

<u>Barrel Roll (2)</u>

In order to decrease the altitude loss due to gravitational effects, this routine assumes that the net acceleration vector $\overline{a}_C$ rotates about the velocity vector $\overline{V}$ at a constant specified rate $\dot{\psi}'$. The net accelerations in the horizontal and vertical planes are

$$\begin{aligned} a_{Ch} &= - a_C \sin \psi' \\ a_{CV} &= a_C \cos \psi' \end{aligned} \tag{108}$$

where ψ' is given by

$$\psi' = \int \dot{\psi}' \, dt \tag{109}$$

Referring to Fig. 8, we see that the acceleration vector $\overline{a}_C$ will rotate uniformly about the velocity vector $\overline{V}$, but the force F_n will vary with the rotation. F_n is given by

$$F_n = \left(\frac{W}{g}\right)\left[a_{Ch}^2 + (a_{CV} + g \cos \gamma)^2\right]^{\frac{1}{2}} \tag{110}$$

In calling for Barrel Roll (2), it is necessary to specify (1) the number of 360-deg rolls required, (2) the number of g's for $\overline{a}_C$ (not F_n/W), and (3) the angular rate $\dot{\psi}_B$ in degrees per second.

Constant-Roll-Angle Climbing or Diving Turn (GRFØLL)

The term roll angle, as used here, is an angle referenced to the vehicle's pitch, roll, and yaw coordinate system, as distinguished from bank angle ψ_B, which is referenced to a wind or velocity system.* In TACTICS, the roll angle ψ is taken to be the angle between the vehicle pitch axis, i.e., the wings of the vehicle, and a line through the c.g. of the vehicle both normal to the longitudinal (i.e., roll) axis and parallel to the horizontal plane. The constant-roll-angle maneuver requires this angle to be held constant. The normal force in g's, F_n/W, is also specified. For values of ψ between -90 deg to -180 deg and between +90 deg to +180 deg, a diving turn to the left or right, respectively, will result.

Since F_n and ψ (see Appendix A for definition) are specified, the subroutine solves the problem of finding the components a_{oh} and a_{oV} to force the following two conditions:

$$a_{oh}^2 + \left(a_{oV} + g \cos \gamma_V\right)^2 = \left[\left(\frac{g}{W}\right)F_n\right]^2 = \text{constant} \tag{111}$$

and

$$\psi = \psi(a_{oh}, a_{oV}) = \text{constant} \tag{112}$$

The functional relationship $\psi(a_{oh}, a_{oV})$ is not amenable to solving these two equations for a_{oh} and a_{oV} directly, so numerical methods are used. The technique employed is to rotate the lift vector $\overline{L}$ about $\overline{V}$ by an amount $\Delta\mathscr{E}$ (the absolute value $|\overline{L}|$ is known but the direction is not). With each incremental rotation $\Delta\mathscr{E}$, the angle ψ is computed until the required conditions are fulfilled within an error $\Delta\psi$ less than 1 mrad. Note that by holding a roll angle of absolute magnitude greater than zero, the vehicle will continuously be climbing or diving until the *longitudinal roll axis* becomes vertical. However, the velocity vector will be

*Apparently, there is no universal agreement in texts or among engineers on terminology or exact definition.

lagging by approximately the angle of attack α. When this condition is reached, roll angle becomes singular and a warning message is printed. Thereafter, the vehicle will continue its climb (or dive) until the velocity vector becomes vertical; then straight flight up (or down) is assumed.

Ballistic 20-mm Projectile

This routine simulates the trajectory of a 20-mm type M56A1 projectile. The drag coefficient C_D versus Mach number characteristics are represented by several linear functions over various Mach-number regimes. The commanded accelerations correspond to a 0-g ballistic trajectory:

$$a_h = 0.0$$

$$a_V = - g \cos \gamma \tag{113}$$

$$F_n = 0.0$$

The projectile weight is 0.22 lb, and the reference area is 0.0033842 ft^2. The muzzle velocity ranges from 3350 to 3450 ft/sec and should be set as ΔV in the launch subroutine (see Launch above).

CLOSED-LOOP CONTROL LAWS

Proportional Navigation

The commanded lateral acceleration $\bar{a}_C$ is proportional to the space rate of rotation of the LOS between missile and target. Expressed in vector notation,

$$\bar{a}_C = \lambda V \bar{\omega}_r \times \bar{1}_v \tag{114}$$

where

λ = the "navigation constant" (it may be treated as either a constant or a variable)

V = missile speed

$\overline{\omega}_r$ = relative angular-rate vector as defined by Eq. (5) in Section III.

$\overline{1}_v$ = unit vector along the missile velocity $\overline{V}$, i.e., $\overline{1}_V = \overline{V}/V$

The direction of the acceleration may be defined by

$$\overline{1}_\perp = \overline{\omega}_r \times \overline{1}_V/\omega_r \tag{115}$$

The commanded acceleration $\overline{a}_C$ may be resolved into horizontal and vertical components by

$$\begin{aligned} \overline{a}_{Ch} &= a_C\left(\overline{1}_\perp \cdot \overline{1}_A\right) \\ \overline{a}_{CV} &= a_C\left(\overline{1}_\perp \cdot \overline{1}_D\right) \end{aligned} \tag{116}$$

Biased Proportional Navigation

The commanded lateral acceleration $\overline{a}_C$ is proportional to a biased (or "modified") space rate of rotation of the LOS between interceptor and target. The bias term may be either a constant or a time variable; and its general purpose is (1) to provide for accelerations involved in the problem and/or (2) to shape (i.e., straighten or curve) the resultant trajectory. The calculations to determine this quantity may range in complexity from the estimation of a constant to the solution of complicated second-order prediction equations. The present version of this routine uses a bias term to account for an average boost velocity increment, such as would occur for a high-specific-impulse missile launch. It may be used, for example, in simulating a lead collision course where the interceptor aircraft is to steer so as to account for the ΔV occurring at missile launch. The equations used are

$$\overline{a}_C = \lambda V\left(\overline{\omega}_r - \overline{\omega}_B\right) \times \overline{1}_V \tag{117}$$

where

$$\overline{\omega}_B = \frac{\left(\Delta V\ \overline{r}_{ij} \times \overline{1}_V\right)}{r_{ij}^2} \tag{118}$$

ΔV = terminal boost velocity

$\overline{r}_{ij}$ = relative interceptor-target range vector

$\overline{1}_V$ = unit vector along the interceptor velocity $\overline{V}$ $\left(\overline{1}_V = \overline{V}/V\right)$

Note that the bias-term evaluation described above requires an a priori knowledge of only ΔV and LOS orientation and hence requires minimal hardware on board the vehicle.

Lead Collision

This guidance law may be considered as a more sophisticated version of biased proportional navigation. Since range and range-rate information are assumed available, it is feasible to make more accurate predictions in providing for the missile launch. Moreover, the effective gain, λ, may be varied as a function of a computed time-to-go, T. The miss vector (lead collision) form of guidance is briefly described in the following paragraphs. Consider the following miss vector diagram (the vectors are not necessarily co-planar), in which

T = time-to-go, e.g., from "now" to the missile launching point

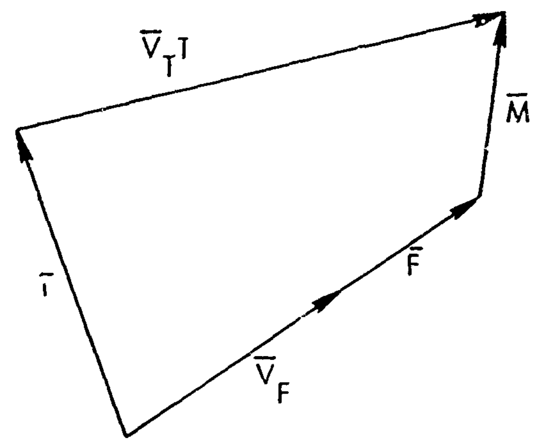
$\overline{M}$ = a miss vector--to be driven to zero.

$\overline{F}$ = vector representing the distance and direction a missile will travel after launch in a flight time t_f

$\overline{V}_F$ = launching aircraft velocity vector

$\overline{V}_T$ = target velocity vector

$\overline{r} = \overline{r}_{ij}$, the relative range vector (LOS)



With reference to the vector diagram,

$$\overline{V}_T T + \overline{r} = \overline{V}_F \left(T - t_f\right) + \overline{F} + \overline{M} \tag{119}$$

from which

$$\overline{M} = \overline{r} - \overline{F} + \overline{V}_F t_F + \left(\overline{V}_T - \overline{V}_F\right) T \tag{120}$$

Next, calculate the vector $\overline{F}$ and the flight time t_f. Designate r_o as some preset value of interceptor-target range at which the missile is to be launched. When $r = r_o$ at time of launch,

$$\overline{r}_o = \overline{r}$$

$$T = t_f$$

$$\left(V_F + \Delta V\right)^2 t_f^2 = r_o^2 + 2\overline{r}_o \cdot \overline{V}_T t_f + V_T^2 \, t_f^2 \tag{121}$$

From this quadratic equation t_f may be determined. Then

$$\overline{F} = \Delta V t_f \, \overline{1}_V \tag{122}$$

The vector $\overline{M}$ may now be resolved into components parallel and perpendicular to the LOS vector $\overline{r}$. First, consider the parallel component, designated $M_{||}$:

$$M_{||} = \overline{M} \cdot \overline{1}_r = r - \overline{F} \cdot \overline{1}_r + \overline{V}_F \cdot \overline{1}_r + \dot{r}\, T \tag{123}$$

since

$$\dot{r} = \left(\overline{V}_T - \overline{V}_F\right) \cdot \overline{1}_r \tag{124}$$

Next arbitrarily set $M_{||} = 0$ and solve for T, the time-to-go.

$$T = \frac{\left(\overline{F} - \overline{V}_F\right) \cdot \overline{1}_r - r}{\dot{r}} = \frac{t_f \Delta\overline{V} \cdot \overline{1}_r - r}{\dot{r}} \tag{125}$$

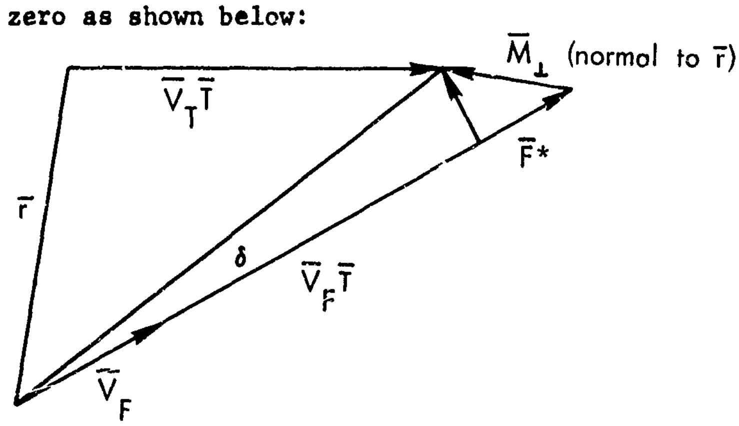
since

$$F = \left(V_F + \Delta V\right) t_f \tag{126}$$

For notational convenience we will define

$$\overline{F}^* = \Delta V t_f \, \overline{1}_V \tag{127}$$

Because of the arbitrary definition of T in Eq. (125), the parallel component of miss $M_{||}$ is zero at T. The problem is reduced to finding an expression for interceptor lateral acceleration so that $M_\perp$ at time T is driven to zero as shown below:



Note that the lateral acceleration is by definition normal to $\overline{V}_F$. The lateral acceleration command may (arbitrarily) be made proportional to M_1

$$a_C \sim \delta \tag{128}$$

where

$$\delta = \frac{M_1}{V_F T + F^*}$$

$$a_C = \lambda V \delta$$

where λ represents the guidance-loop gain. The magnitude and orientation of $\overline{a}_C$ is determined as follows:

$$M_1 = \left|\overline{M} \times \overline{1}_r\right| = \left|-\overline{F}^* \times \overline{1}_r + T\left(\overline{V}_T - \overline{V}_F\right) \times \overline{1}_r\right| = \left|-\overline{F}^* \times \overline{1}_r + \overline{\omega} rT\right| \tag{129}$$

The orientation of the acceleration $\overline{a}_C$ is given by

$$\overline{1}_1 = \overline{a}_C / a_C = \frac{\left(\overline{M} \times \overline{1}_r\right) \times \overline{1}_V}{M_1} \tag{130}$$

The complete expression for $\overline{a}_C$ may now be written as

$$\overline{a}_C = \frac{\lambda V_F}{V_F T + F^*}\left[-\left(\overline{F}^* \times \overline{1}_r\right) + \overline{\omega}\, rT\right] \times \overline{1}_V \tag{131}$$

This relationship may be expressed in a different form to show the analogy with biased proportional navigation:

$$\overline{a}_C = \lambda' V\left(\overline{\omega} - \overline{\omega}_B\right) \times \overline{1}_V \tag{132}$$

where

$$\lambda' = \frac{\lambda r T}{V_F T + F^*} \qquad (133)$$

$$\bar{\omega}_B = \frac{\bar{F}^* \times \bar{1}_r}{rT} \qquad (134)$$

Note that λ' and $\bar{\omega}_B$ are time-varying functions.

Pure Pursuit Course (1)

The commanded lateral acceleration $\bar{a}_C$ is proportional to the angular difference $\Delta\gamma$ between the interceptor's velocity vector $\bar{V}$ and the interceptor-target range vector $\bar{r}_{ij}$. This may be expressed as follows:

$$\Delta\gamma = \sin^{-1}|\bar{1}_V \times \bar{1}_r|$$

$$\bar{1}_1 = \left(\bar{1}_V \times \bar{1}_r\right) \times \bar{1}_V/\Delta\gamma$$

$$\bar{a}_C = \lambda V \Delta\gamma \bar{1}_1 \qquad (135)$$

$$a_{Ch} = a_C\left(\bar{1}_1 \cdot \bar{1}_A\right)$$

$$a_{CV} = a_C\left(\bar{1}_1 \cdot \bar{1}_D\right)$$

It is significant to mention that a pursuit-course navigation law is not usually applied for missile terminal homing guidance, since $|\bar{a}_C|$ may become infinitely large as :he relative range approaches zero. However, this control law is very useful for describing fighter aircraft "gunsight aiming" flight paths. The gain constant λ is arbitrarily selected, since it represents the pilot's reaction and skill in keeping the target in the crceshairs (practical values range from about 4 to 40).

Pursuit (2)

Pure pursuit-course navigation is defined in terms of maintaining the velocity vector of the pursuing vehicle along the LOS to the target.

In practice, a pilot would only be able to approximate a pure pursuit course without angle of attack or velocity information. Rather than the velocity vector, the most convenient reference is the airframe itself, i.e., the longitudinal axis. With modern jet aircraft in high-g maneuvers, angle-of-attack differences may range from 15 to 20 deg. The Pursuit (2) routine may be termed a deviated pursuit course where the bearing angle B between the longitudinal axis of the pursuing vehicle and the LOS is maintained at a near-zero value. A necessary condition for beginning the deviated pursuit guidance is that the magnitude of the bearing angle be less than α_{max}, which corresponds to the C_{Lmax} condition. The routine calls for pure pursuit guidance until

$$|B| < \alpha_{max}$$

The trajectory we wish to simulate is described mathematically as follows: Find an a_h and an a_V resulting in a net acceleration vector $\overline{a}$ and a normal force vector $\overline{F}_n$. This normal force will require a lift vector $\overline{L}$ with a corresponding angle of attack α. The problem requires solving for the three unknown components of $\overline{F}_n$, all functions of C_L and α. There are two functions which should be minimized:

$$f_1\left(a_h, a_V\right) = \alpha - \gamma \tag{136}$$

where γ is the angle between $\overline{V}$ and the LOS and α is the angle between $\overline{V}$ and the longitudinal axis $\overline{1}_T$, and

$$f_2\left(a_h, a_V\right) = |B| \tag{137}$$

In other words, we would like $\alpha = \gamma$ and $|B| = 0$ for an idealized trajectory. Since nonlinear relationships (i.e., table values) and transcendental relationships are involved, numerical iterative techniques are used to determine the acceleration $\overline{a}$ and its components a_h and a_V. As a first approximation, the proportional navigation law is used to obtain a first guess for $\overline{1}_1$. Next, the $C_L - \alpha$ correspondence is established for $\alpha = \gamma$, thereby determining L and $\overline{F}_n$. Computations

are then performed to obtain a_h, a_V, and ultimately B. The vector $\overline{F}_n$ is then rotated by a small angle $\mathcal{E}$ to minimize $|B|$.

On-Off (1)

This guidance law (sometimes called bang-bang) is analogous to proportional navigation. The direction of the lateral acceleration unit vector $\overline{I}_\perp$ is determined identically. However, the magnitude of the acceleration is not proportional to the angular rotation rate $|\overline{\omega}_r|$ of the LOS, but dual-valued. That is,

$$a = 0.0 \quad \text{when } 0 < |\overline{\omega}_r| < \mathcal{E}$$
$$a = a_{Smax} \quad \text{when } |\overline{\omega}_r| > \mathcal{E} \tag{138}$$

where $\mathcal{E}$ is some small specified threshold value of angular rate (e.g., 1 mrad/sec) required for stability.

On-Off (2)

This guidance law is analogous to pure pursuit-course navigation, and the direction of $\overline{I}_\perp$ is determined identically. As in Eq. (138),

$$a = 0.0 \quad \text{when } 0 < |\gamma| < \mathcal{E}$$

$$a = a_{Smax} \quad \text{when } |\gamma| > \mathcal{E} \tag{139}$$

where $\mathcal{E}$ is some small specified threshold value of angle (e.g., 1 mrad) required for stability.

Missile (X)

The commanded lateral acceleration $\overline{a}_C$ is proportional to the space rate of rotation of the LOS between interceptor and target, i.e., proportional navigation. Many of the significant guidance and aerodynamic parameters vary with Mach number and are unique to this hypothetical missile design, as are the boost/burn/guide-time intervals. Accordingly,

these detailed characteristics are packaged into a specialized subroutine. The pertinent characteristics associated with this missile performance are listed below.

Constants

$A = 0.13635\ ft^2$

W_o (initial weight) = 187 lb

W_B (burnout weight) = 125 lb

t_B (burn time) = 4.75 sec

$\dot{W} = (125 - 187)/4.75 = -13.06$ lb/sec

Thrust = 3053 lb

Time Constants

$\tau_1 = 0.1$

$\tau_2 = 0.15$

$\tau_3 = 0.15\ (\rho_o/\rho)$

Parameters

$\lambda = 0.35 + 12.22/M^{3/2}$

$$a_{Smax} = \min \frac{15.0M\ (\rho_o/\rho)}{0.95},\quad (16.52 - 1.52\ M) \qquad (M = \text{Mach number})$$

$C_D = C_{D_o} + dC_D/d(C_L^2)$

The values of $dC_L/d\alpha$ and $dC_D/d(C_L^2)$ with respect to Mach number are given in the tables on the following page:

$dC_L/d\alpha$ (per rad)	Mach No.
$23.52 + 109.9/M^2$	≥ 1.85
55.5	$\geq 1.5, < 1.85$
$12 + 45M$	$\geq 1.1, < 1.5$
37.5	< 1.1
$0.51 + 2.52/M^2$	≥ 1.5
2.19	$\geq 1.0, < 1.5$
$0.515 + 1.675M$	$\geq 0.6, < 1.0$
1.5	< 0.6

$dC_D/d(C_L^2)$ (per rad)	Mach No.
$10 + 93/M$	≥ 1.5
72	< 1.5
$24.33 + 43.33M$	$\geq 0.5, < 1.1$
46	< 0.5

Appendix D

INSTRUCTIONS FOR CALLING OPTIONAL SUBROUTINES

Many optional subroutines are available with TACTICS, including maneuver and guidance routines. (Launch and captive-flight routines are considered special cases in this category, as shown in Table 2.)

When a maneuver or guidance subroutine is called, the following three arguments must always be specified (unless the routines are formulated to represent specific vehicles such as MISILX):

I: Vehicle to be used (1, 2, or 3).

IAERØ: Type of aerodynamics to be used.
- IAERØ = 1, analytic functions.
- IAERØ = 2, aerodynamic tables.
- IAERØ = 3, VDØT = 0.0.

ITHR: Thrust to be used (lb).
- ITHR = 1, afterburner thrust obtained from tables.
- ITHR = 2, military thrust obtained from tables.
- ITHR = 3, afterburner thrust = constant (DATA 96, 101, 106).
- ITHR = 4, military thrust = constant (DATA 95, 100, 105).
- ITHR = 5, thrust = 0.0.

In addition, the following arguments may be required:

DELV: Boost velocity of missile (ft/sec).

MØDE: Flag indicating captive-flight option.
- MØDE = 1, holds missile in captive flight on fighter.
- MØDE = 2, sets all quantities related to designated vehicle equal to zero.
- MØDE = 3, holds missile in captive flight on target.

EPSLØN: Threshold value used in on-off control laws for stability.

LEVEL(I): Flag set in subroutine STRLVL used to communicate to PØLICY that the vehicle velocity vector is in a horizontal plane within a tolerance of 0.002 rad.

GFØRC: Total lateral acceleration, including gravitational effects, for a maneuver (g's).

Table 2

MANEUVER AND GUIDANCE ROUTINES

Routine and Argument Listing	Function
BIASPN(I, DELV, IAERØ, ITHR)	Biased proportional navigation guidance law
BRLRL1(I, RØLLS, IRØLL, GFØRC, RØLLRT, IAERØ, ITHR)	Constant-g barrel roll where the force $\overline{F}_n$ rotates about velocity vector
BRLRL2(I, RØLLS, IRØLL, GFØRC, RØLLRT, IAERØ, ITHR)	Constant-g barrel roll where the lateral acceleration $\overline{a}_o$ rotates about the velocity vector
B20MM(I)	Simulating ballistics for 20-mm cannon projectiles
CAPFLT(I, MØDE)	Captive flight
CLIMB1(I, GFØRC, IAERØ, ITHR)	Simple constant-g climb
CLIMB2[a](I, IAERØ, ITHR)	Constant-Mach climb depending on available excess thrust
DIVE1(I, GFØRC, IAERØ, ITHR)	Simple constant-g dive
GFRØLL(I, GFØRC, RØLLL, IAERØ, ITHR)	Desired GFØRCE AND RØLL (GFØRC and RØLLL in argument listing) are specified and corresponding accelerations are computed
LAUNCH(I, DELV, IAERØ, ITHR)	Launches missile
LEADCL(I, DELV, IAERØ, ITHR)	Lead collision guidance law
LTRN1(I, GFØRC, IAERØ, ITHR)	Simple constant-g left turn
LTRN2[a](I, IAERØ, ITHR)	Constant-altitude, constant-Mach left turn depending on available excess thrust
LTRN3[a](I, GFØRC, IAERØ, ITHR)	Constant-Mach left turn where GFØRC is specified
LTRN4[a](I, GFØRC, IAERØ, ITHR)	Constant-Mach left turn unless aerodynamic flight conditions (CLMAX) prohibit obtaining specified value, in which case aircraft will maneuver at maximum acceleration in conformity with CLMAX limitations, keeping Mach number constant

[a]These subroutines require the routine MACHR8.

Table 2 (continued)

Routine and Argument Listing	Function
LTRN5[a](I, GFØRC, RØLLL, IAERØ, ITHR)	Constant-g climbing or diving left turn as defined by bank angle RØLLL (see Appendix C)
MISILX(I)	Proportional navigation missile
MISIL2(I)	Rearward-launched MISILX(I)
ØNØFF(I, EPSLØN, IAERØ, ITHR)	Bang-bang (on-off) control law, analogous to PRØNAV
ØNØFF2(I, EPSLØN, IAERØ, ITHR)	Bang-bang (on-off) control law, analogous to PRSUIT
PRØNAV(I, IAERØ, ITHR)	Proportional-navigation guidance law
PRSUIT(I, IAERØ, ITHR)	Pursuit-course navigation, velocity vector pointed down LOS
PRSUIT2(I, IAERØ, ITHR)	Pursuit-course navigation, thrust vector pointed down LOS: must be used with subroutine FUNCTN(I) (See Appendix C)
RTRN1(I, GFØRC, IAERØ, ITHR)	Simple constant-g right turn
RTRN2[a](I, IAERØ, ITHR)	Constant-altitude, constant-Mach right turn depending on available excess thrust
RTRN3[a](I, GFØRC, IAERØ, ITHR)	Constant-Mach right turn where GFØRC is specified
RTRN4[a](I, GFØRC, IAERØ, ITHR)	Constant-Mach right turn unless aerodynamic flight conditions (CLMAX) prohibit obtaining specified value, in which case aircraft will maneuver at maximum acceleration in conformity with CLMAX limitations, keeping Mach number constant
RTRN5[a](I, GFØRC, RØLLL, IAERØ, ITHR)	Constant-g climbing or diving right turn as defined by bank angle RØLLL (see Appendix C)
STRFLT(I, IAERØ, ITHR)	Straight flight
STRLVL(I, IAERØ, ITHR, LEVEL)	Level off to horizontal position

[a]These subroutines require the routine MACHR8.

Appendix E

DESCRIPTION OF INPUT DATA*

Data No.	Program Variable	Symbol	Description
1	IRF		Position flag, vehicle 1 (Cartesian coordinates); IRF = 0
2	R(1,1)	x_1	x-coordinate, vehicle 1
3	R(1,2)	y_1	y-coordinate, vehicle 1
4	R(1,3)	z_1	z-coordinate, vehicle 1
5	WO(1)	W_{01}	Initial weight, vehicle 1
6	IRF		Position flag, vehicle 1 (spherical coordinates); IRF = 1
7	R(1,4)	r_1	Range vector magnitude, vehicle 1
8	R(1,5)	θ_1	Range angle measured in horizontal plane, vehicle 1 (see Fig. 3)
9	R(1,6)	φ_1	Range angle measured in vertical plane, vehicle 1 (see Fig. 3)
10	AREA(1)	A_1	Reference area, vehicle 1
11	IVF		Velocity flag, vehicle 1 (Cartesian coordinates); IVF = 0
12	V(1,1)	$\dot{x}_1$	x-component of velocity, vehicle 1
13	V(1,2)	$\dot{y}_1$	y-component of velocity, vehicle 1
14	V(1,3)	$\dot{z}_1$	z-component of velocity, vehicle 1
15	ASMAX(1)	a_{Smax1}	Maximum lateral acceleration limit (structural), vehicle 1
16	IVF		Velocity flag, vehicle 1 (spherical coordinates); IVF = 1 Magnitude only (calls AIM); IVF = 2
17	V(1,4)	V_1	Velocity vector magnitude, vehicle 1
18	V(1,5)	θ_{V1}	Velocity angle measured in horizontal plane, vehicle 1 (see Fig. 4)
19	V(1,6)	γ_1	Flight-path angle measured in vertical plane, vehicle 1 (see Fig. 4)

*Units are distance (ft), time (sec), velocity (ft/sec or Mach no.), acceleration (g's), angles (deg), weight (lb), area (ft^2).

Data No.	Program Variable	Symbol	Description
20	JATMØS		Flag for reading in velocity magnitudes (DATA 17, 33, and 56) JATMØS = 0 for ft/sec JATMØS = 1 for Mach number (applies to all vehicles)
21	TGUIDE(1)	t_{g1}	Time interval that vehicle 1 (if a missile) is to fly unguided after launch
22			
23			
24			
25	WO(2)	W_{02}	Initial weight, vehicle 2
26	AREA(2)	A_2	Reference area, vehicle 2
27	ASMAX(2)	a_{Smax2}	Maximum lateral acceleration limit (structural), vehicle 2
28	ICAP		Flag to indicate initial-condition flight status of vehicle 2 (see Appendix H)
29	RMTMAX		Maximum flight range, vehicle 2
30	R(2,1)	x_2	x-coordinate, vehicle 2
31	R(2,2)	y_2	y-coordinate, vehicle 2
32	R(2,3)	z_2	z-coordinate, vehicle 2
33	V(2,4)	V_2	Velocity vector magnitude, vehicle 2
34	V(2,5)	θ_{V2}	Angle measured in horizontal plane, vehicle 2 (see Fig. 4)
35	V(2,6)	γ_2	Flight path angle measured in vertical plane, vehicle 2 (see Fig. 4)
36	TBURN1	t_{B1}	First stage rocket motor burning time (may be used for vehicle 1, 2, or 3)
37	TBURN2	t_{B2}	Second stage rocket motor burning time (may be used for vehicle 1, 2, or 3)
38	KLAUN		Decimal fraction of vehicle's maximum range at which it is to be launched (may be used for vehicle 1, 2, or 3)

Data No.	Program Variable	Symbol	Description
39	TGUIDE(2)	t_{g2}	Time interval that vehicle 2 (if a missile) is to fly unguided after launch
40	IRT		Position flag, vehicle 3 (Cartesian coordinates); IRT = 0
41	R(3,1)	x_3	x-coordinate, vehicle 3
42	R(3,2)	y_3	y-coordinate, vehicle 3
43	R(3,3)	z_3	z-coordinate, vehicle 3
44	WO(3)	W_{03}	Initial weight, vehicle 3
45	IRT		Position flag, vehicle 3 (spherical coordinates); IRT = 1
46	R(3,4)	r_3	Range vector magnitude, vehicle 3
47	R(3,5)	θ_3	Range angle measured in horizontal plane, vehicle 3 (see Fig. 3)
48	R(3,6)	φ_3	Range angle measured in vertical plane, vehicle 3 (see Fig. 3)
49	AREA(3)	A_3	Reference area, vehicle 3
50	IVT		Velocity flag, vehicle 3 (Cartesian coordinates); IVT = 0
51	V(3,1)	$\dot{x}_3$	x-component of velocity, vehicle 3
52	V(3,2)	$\dot{y}_3$	y-component of velocity, vehicle 3
53	V(3,3)	$\dot{z}_3$	z-component of velocity, vehicle 3
54	ASMAX(3)	a_{Smax3}	Maximum lateral acceleration limit (structural), vehicle 3
55	IVT		Velocity flag, vehicle 3 (spherical coordinates); IVT = 1 Magnitude only (calls AIM); IRF = 2
56	V(3,4)	V_3	Velocity-vector magnitude, vehicle 3
57	V(3,5)	θ_{V3}	Velocity angle measured in horizontal plane, vehicle 3 (see Fig. 4)
58	V(3,6)	γ_3	Flight-path angle measured in vertical plane, vehicle 3 (see Fig. 4)
59	TGUIDE(3)	t_{g3}	Time interval that vehicle 3 (if a missile) is to fly unguided after launch

Data No.	Program Variable	Symbol	Description
60			(Not used)
61			(Not used)
62	TIME	t_o	Running time (when entered as input data, it is equivalent to initial time, i.e., time at which problem is to start)
63	DTPØ		Print interval, i.e., time increment for printout
64	TØTAL		Time limit placed on internal program running time if it is not to be terminated after miss calculation
65	ITAU(1)		Flag indicating number of first-order time lags, vehicle 1 (see Section V)
66	TAU(1,1)	τ_{11}	First time lag, vehicle 1
67	TAU(1,2)	τ_{12}	Second time lag, vehicle 1
68	TAU(1,3)	τ_{13}	Third time lag, vehicle 1
69	LAMDAO(1)	λ_{01}	Navigation constant for guidance, vehicle 1
70	ITAU(2)		Flag indicating number of first-order time lags, vehicle 2 (see Section V)
71	TAU(2,1)	τ_{21}	First time lag, vehicle 2
72	TAU(2,2)	τ_{22}	Second time lag, vehicle 2
73	TAU(2,3)	τ_{23}	Third time lag, vehicle 2
74	LAMDAO(2)	λ_{02}	Navigation constant for guidance, vehicle 2
75	ITAU(3)		Flag indicating number of first-order time lags, vehicle 3 (see Section V)
76	TAU(3,1)	τ_{31}	First time lag, vehicle 3
77	TAU(3,2)	τ_{32}	Second time lag, vehicle 3
78	TAU(3,3)	τ_{33}	Third time lag, vehicle 3
79	LAMDAO(3)	λ_{03}	Navigation constant for guidance, vehicle 3

Data No.*	Problem Variable	Symbol	Description
80	CLMAX(1)	C_{Lmax1}	Maximum aerodynamic lift coefficient
81	CDOCØN(1)	C_{D01}	Zero or profile drag coefficient: C_{D_o}, vehicle 1
82	BCØN(1)	dC_D/dC^2_{L1}	Coefficient used with parabolic approximation for drag coefficient as a function of lift coefficient, vehicle 1
83	SLØPE(1)	$dC_L/d\alpha_1$	Slope of C_L vs α curve, assumed to be a constant (analytic functions), vehicle 1
84	ALPHAO(1)	α_{01}	Total zero-lift angle of attack, vehicle 1
85	CLMAX(2)	C_{Lmax2}	Maximum aerodynamic lift coefficient, vehicle 2
86	CDOCØN(2)	C_{D02}	Zero or profile drag coefficient: C_{D_o}, vehicle 2
87	BCØN(2)	dC_D/dC^2_{L2}	Coefficient used with parabolic approximation for drag coefficient as a function of lift coefficient, vehicle 2
88	SLØPE(2)	$dC_L/d\alpha_2$	Slope of C_L vs α curve, assumed to be a constant, vehicle 2
89	ALPHAO(2)	α_{02}	Total zero-lift angle of attack, vehicle 2
90	CLMAX(3)	C_{Lmax3}	Maximum aerodynamic lift coefficient, vehicle 3
91	CDOCØN(3)	C_{D03}	Zero or profile drag coefficient, vehicle 3
92	BCØN(3)	dC_D/dC^2_{L3}	Coefficient used with parabolic approximation for drag coefficient as a function of lift coefficient, vehicle 3
93	SLØPE(3)	$dC_L/d\alpha_3$	Slope of C_L vs α curve, assumed to be a constant, vehicle 3
94	ALPHAO(3)	α_{03}	Zero-lift angle of attack, vehicle 3

*Note that DATA 80-109 are required only when analytic functions are used instead of table values.

Data No.*	Program Variable	Symbol	Description
95	THCØN(1)	T_{M1}	Constant for military thrust, vehicle 1
96	TABCØN(1)	T_{ab1}	Constant for afterburner thrust, vehicle 1
97	IMPLSE(1)	I_1	Specific impulse of rocket motor, vehicle 1
98	WBURN(1)	W_{B1}	Weight at rocket motor burnout, vehicle 1
99	ABØØST(1)	a_{b1}	Boost acceleration (assumed to be a constant), vehicle 1
100	THCØN(2)	T_{M2}	Constant for military thrust, vehicle 2
101	TABCØN(2)	T_{ab2}	Constant for afterburner thrust, vehicle 2
102	IMPLSE(2)	I_2	Specific impulse of rocket motor, vehicle 2
103	WBURN(2)	W_{B2}	Weight at rocket motor burnout, vehicle 2
104	ABØØST(2)	a_{b2}	Boost acceleration (assumed to be a constant), vehicle 2
105	THCØN(3)	T_{M3}	Constant for military thrust, vehicle 3
106	TABCØN(3)	T_{ab3}	Constant for afterburner thrust, vehicle 3
107	IMPLSE(3)	I_3	Specific impulse of rocket motor, vehicle 3
108	WBURN(3)	W_{B3}	Weight at rocket motor burnout, vehicle 3
109	ABØØST(3)	a_{b3}	Boost acceleration (assumed to be a constant), vehicle 3
110	KINTEG		Flat- or round-earth coordinate system flag Flat-earth: KINTEG = 0 Round-earth: KINTEG = 1
111	ALT(1)	h_1	Altitude, vehicle 1 (geocentric coordinates)

*Note that DATA 80-109 are required only when analytic functions are used instead of table values.

Data No.	Program Variable	Symbol	Description
112	LØNG(1)	Λ_1	Longitude, vehicle 1 (geocentric coordinates)
113	LAT(1)	Φ_1	Latitude, vehicle 1 (geocentric coordinates)
114	INERF		Flag for reading vehicle 1 velocity in local or inertial system Local: INERF = 0 Inertial: INERF = 1
115	IRØT8		Flag for earth's rotation Nonrotating: IRØT8 = 0 Rotating: IRØT8 = 1
116	ALT(3)	h_3	Altitude, vehicle 3 (geocentric coordinates)
117	LØNG(3)	Λ_3	Longitude, vehicle 3 (geocentric coordinates)
118	LAT(3)	Φ_3	Latitude, vehicle 3 (geocentric coordinates)
119	INERT		Flag for reading vehicle 3 velocity in local or inertial system Local: INERT = 0 Inertial: INERT = 1
120	LØNGO	Λ_0	Longitude of the local coordinate system origin
121	LATO	Φ_0	Latitude of the local coordinate system origin
122	JINTEG		Integration flag Variable-step Adams-Moulton: JINTEG = 0 Fixed-step Runge-Kutta: JINTEG = 1 Fixed-step Adams-Moulton: JINTEG = 2 Variable step with exact printout: JINTEG = 3 (see Section XIII)
123	ERTEST		Number of significant digits of accuracy required (see Section XIII) of numerical integration
124	MINMR	R_{MIN}	Vehicle 2 range to target within which program will automatically initiate process for miss-distance computation
125	DVTH(1)	$\Delta\theta_1$	Assumed error in θ_V for aiming vehicle 1

Data No.	Program Variable	Symbol	Description
126	DVTH(2)	$\Delta\theta_2$	Assumed error in θ_V for aiming vehicle 2
127	DVTH(3)	$\Delta\theta_3$	Assumed error in θ_V for aiming vehicle 3
128	DVPHI(1)	$\Delta\gamma_1$	Assumed error in γ_1 for aiming vehicle 1
129	DVPHI(2)	$\Delta\gamma_2$	Assumed error in γ_2 for aiming vehicle 2
130	DVPHI(3)	$\Delta\gamma_3$	Assumed error in γ_3 for aiming vehicle 3
131	BETA(1)	β_1	Ballistic coefficient, vehicle 1
132	BETA(2)	β_2	Ballistic coefficient, vehicle 2
133	BETA(3)	β_3	Ballistic coefficient, vehicle 3
134	DTMIN		Minimum integration step size for backup to compute miss distance
135	HMIN		Minimum allowable size of integration step used in variable Adams-Moulton mode integration (excluding miss distance computation)
136	DTØ		Initial (starting) value of integration step size
137	HMX		Maximum specified value for integration step size used in variable Adams-Moulton mode integration
138	ELEVMX(1)	ε_{max1}	Maximum elevation angle limit, vehicle 1
139	ELEVMX(2)	ε_{max2}	Maximum elevation angle limit, vehicle 2
140	ELEVMX(3)	ε_{max3}	Maximum elevation angle limit, vehicle 3
141	AZMAX(1)	η_{max1}	Maximum azimuth angle limit, vehicle 1
142	AZMAX(2)	η_{max2}	Maximum azimuth angle limit, vehicle 2
143	AZMAX(3)	η_{max3}	Maximum azimuth angle limit, vehicle 3

Appendix F

AERODYNAMIC AND PROPULSION TABLES

Tables furnishing variables for computing the aerodynamic functions and the propulsion characteristics of specific aircraft, including the F-104 and F-105, are available. These are to be used when IAERØ = 2 (when the table section of AERØDN is being used).

An example set of tables for an aircraft is given below; sets of tables for other aircraft may be constructed following a similar format.

- o Maximum lift coefficient as a function of Mach number.
- o Drag coefficient as a function of Mach number and lift coefficient.
- o ΔC_D, change in C_D when external stores, e.g., fuel tanks, are jettisoned (the program does not now have the feature for using ΔC_D, but cards must be included in data deck).
- o Afterburner thrust as a function of Mach number and altitude.
- o Fuel flow (afterburner thrust) as a function of Mach number and altitude.
- o Military thrust as a function of Mach number and altitude.
- o Fuel flow (military thrust) as a function of Mach number and altitude.
- o Placard limit (maximum permissible Mach number) as a function of altitude.
- o Angle of attack as a function of Mach number and lift coefficient.

See Table 3 for the format of an aerodynamic table using the above-mentioned functions. Figure 35 shows the actual data deck for the F-104 aircraft.

Table 3

FORMATS FOR AERODYNAMIC TABLES

Table	No. of Cards	Card No.	Quantity	Format	No. of Values
C_{Lmax}	3	1-3	Mach number	18F4.3	42
	3	4-6	C_{Lmax}	18F4.3	42
C_D	1	7	Mach number	18F4.3	16
	2	8-9	C_L	18F4.3	32
	32	10-41	C_D	18F4.4	32×16
ΔC_D	9	42-50	ΔC_D	16F4.4	9×16
ABT[a] and fuel flow	1	51	Mach number	18F4.3	16
	2	52-53	Altitude	14F5.0	16
	32	54-85	ABT	14F5.0	16×16
	32	86-117	Fuel Flow	14F5.0	16×16
Military thrust and fuel flow	1	118	Mach number	18F4.3	16
	2	119-120	Altitude	14F5.0	16
	32	121-152	Thrust(military)	14F5.0	16×16
	32	153-184	Fuel Flow	14F5.0	16×16
Placard limit	1	185	Altitude	14F5.4	14
	1	186	Mach number	14F5.4	14
Angle of attack	2	187-188	Mach number	12F6.2	22
	2	189-190	C_L	12F6.2	13
	44	191-234	Alpha	12F6.2	13×22
	211				

[a]Afterburner thrust.

```
C **** F-104 TABLES FCR AERODYNAMICS ****
C
C  MACH, COLIFTMAX
 200 500 600 700 800 850 900 950 97010001050110011501200130014001500l600
1700180019002000220O
 735 737 741 748 766 794 882101010151000 938 911 902 886 840 786 738 697
 654 639 612 591 574
C
C  MACH, CCLIFT, CODRAG
 000 800 850 900 925 950 9751000105011001150120014001600l8002400
 CC0 050 10C 150 200 250 300 350 400 500 600 725 850 95010501150
 161 166 175 197 228 256 310 372 460 669 9411315172620822462 2806
 161 166 175 197 228 256 310 372 460 669 94113151726208224622866
 165 17C 179 199 229 257 314 382 473 684 95813371757212225132931
 170 176 183 205 234 278 336 405 497 719 98913751814220426283087
 191 193 199 218 251 297 363 429 517 7511C221412185922582693 3164
 230 231 243 268 306 354 417 49C 576 80C1081148219382343278 33258
 309 322 338 357 382 428 489 563 656 88111691576203524402878 3350
 379 391 403 420 452 508 576 651 735 98112721691217125993067 3573
 469 472 485 508 554 603 672 757 86811171403183323432809 33263895
 500 504 518 550 598 657 731 821 91911721481193724692950348140 61
 500 508 522 552 599 662 737 830 94112061542202925913094364 54244
 491 500 518 552 602 669 752 848 95812361581208826793212380C4441
 449 457 478 511 572 644 741 852 98212871645217828073379401 14705
 407 415 440 481 543 627 739 857 99813271693227329883658441 55259
 380 [illegible] [illegible] 47C 537 63C 747 87410281388178724213204393947695695
 369 370 394 458 552 678 800 98011401600210030634400574073209140
C
C  CELTA CODRAG
  27  27  29  32  35  39  45  51  51  46  43  30  30  30  30  30
C
C  MACH,ALTITUDE,THRUST, (AFTER-BURNER)
```

Continued

Fig. 35 — F-104 aerodynamic tables

```
 200 600 800 9001000011000120001300014000150001600017000180001900020002200
    0 5000100001500020000250003000035000400004500050000550006000065000

12900112000 9650 8250 7300 6350 5300 4200 3200 2300 1750 1500 1250  750

1535013500128001020 0 8450 7250 6000 4850 3800 2850 2200 1700 1350  800

16750149501320011350 9700 8250 6800 5600 4350 3350 2500 1900 1500  950

17450157501400012200104000 8800 7350 6100 4750 3700 2750 2100 1650 1050

18200165501480013000112500 9600 8100 6650 5250 4000 3050 2300 1750 1200

187501740015650133001220010450 8800 7350 5800 4400 3350 2500 1900 1400

187501830016450148001320011350 9700 8100 6350 4850 3750 2800 2200 1650

1835018300173501580014150123501065 0 8900 7100 5400 4200 3200 2400 1800

17900179001790016900151001330011600 9800 7800 6100 4700 3600 2700 2100

175001750017500175001620014300125001075 0 8600 6700 5200 3900 2900 2300

169001690016900170001660015300134001160 0 9250 7250 5700 4300 3250 2500

164501645016450164001640015800142001240 0 9900 7750 6000 4700 3550 2750

1580015800158001580015800158001455013100105 00 8200 6400 4900 3800 2800

14700147001470014700147001470014550132501070 0 8350 6550 5000 3850 2900

13500135001350013500135001350013500133001070 0 8350 6500 5050 3900 2950

13300133001330013300133001330013300133001070 0 8350 6500 5050 3900 2950

C
C  MACH,ALTITUDE,FUEL FLOW (AFTER-BURNER)
29100251002140018100159001330011500 9400 7900 6800 5700 4700 4000 3700

355  307  265  225  191  160  134  110   90   74   60   50   43   38

396  347  301  257  218  184  153  127  103   83   67   55   46   40

419  360  322  276  235  198  166  137  111   89   73   59   50   42

437  394  334  298  255  214  181  150  121   97   79   64   53   44

449  417  366  320  277  233  197  163  132  106   85   70   57   47

447  439  388  332  297  254  214  178  144  116   93   76   62   51

435  435  410  365  319  275  233  195  157  126  101   82   67   55

419  419  419  388  340  295  253  211  171  137  110   89   72   59

403  403  403  438  361  316  272  232  186  148  119   96   77   63

387  387  387  444  415  336  291  249  200  160  128  103   83   68
```

Continued

Fig. 35 (continued)

```
369  369  369  434  434  390  309  266  214  171  137  110   89   72      1040108
                                                                          1040109
353  353  353  410  410  410  360  282  227  181  145  116   93   76      1040110
                                                                          1040111
337  337  337  370  370  370  370  320  247  196  156  125  101   82      1040112
                                                                          1040113
318  318  318  330  330  330  330  334  271  215  171  137  110   88      1040114
                                                                          1040115
276  276  276  276  276  276  276  346  276  219  175  141  113   90      1040116
                                                                          1040117
C
C  MACH, ALTITUDE, THRUST (MILITARY)
 200 300 400 500 600 700 800 906100011001200150018002000                  1040118
    0 50001000015000200002500030000350004000045000500005500060000065000   1040119
                                                                          1040120
 8140 7240 6200 5300 4500 3800 3100 2550 2000 1540 1140  800  400  100    1040121
                                                                          1040122
 8140 7240 6220 5330 4530 3820 3140 2580 2030 1560 1160  810  410  110    1040123
                                                                          1040124
 8140 7240 6240 5360 4560 3840 3180 2620 2060 1590 1190  820  420  120    1040125
                                                                          1040126
 8140 7250 6270 5390 4590 3860 3220 2660 2090 1620 1220  830  430  130    1040127
                                                                          1040128
 8140 7250 6330 5460 4670 3930 3300 2730 2150 1660 1260  860  460  160    1040129
                                                                          1040130
 8140 7260 6420 5560 4770 4030 3400 2820 2220 1720 1320  900  500  200    1040131
                                                                          1040132
 8150 7370 6590 5760 4970 4230 3580 2980 2350 1820 1390  990  570  240    1040133
                                                                          1040134
 8160 7490 6760 5970 5180 4440 3770 3140 2480 1920 1470 1080  650  280    1040135
                                                                          1040136
 8260 7590 6890 6170 5430 4690 4010 3360 2650 2060 1560 1200  780  310    1040137
                                                                          1040138
 8360 7700 7030 6390 5700 4970 4230 3600 2840 2210 1660 1330  920  340    1040139
                                                                          1040140
 8460 7830 7200 6630 6010 5280 4570 3870 3060 2380 1780 1480 1080  380    1040141
                                                                          1040142
 7600 7600 7600 7450 6450 5840 5250 4590 3660 2860 2160 1860 1460  500    1040143
                                                                          1040144
 6700 6700 6700 6700 6530 6300 5800 4820 3870 3020 2470 2150 1750  800    1040145
                                                                          1040146
 5200 5200 5200 5200 5150 5090 5000 4850 3960 3080 2500 2200 1800  900    1040147
                                                                          1040148
                                                                          1040149
                                                                          1040150
                                                                          1040151
                                                                          1040152
C
C  MACH, ALTITUDE, FUEL FLOW (MILITARY)
 8300 7300 6200 5100 4100 3200 2600 2200 1800 1500 1000  500              1040153
                                                                          1040154
 8400 7400 6300 5200 4400 3400 2800 2300 2000 1600 1100  600              1040155
                                                                          1040156
 8700 7700 6500 5500 4600 3700 3100 2500 2100 1700 1200  700              1040157
                                                                          1040158
 9000 8000 6800 5800 4900 4000 3300 2800 2200 1800 1300  800              1040159
                                                                          1040160
 9500 8300 7100 6100 5200 4300 3600 3000 2400 1900 1400  900              1040161
                                                                          1040162
```

Continued

Fig. 35 (continued)

```
10000 8700 7500 6500 5600 4700 3900 3200 2600 2000 1600 1200                1040163
                                                                            1040164
10500 9100 7900 6900 6000 5000 4300 3500 2800 2200 1700 1300                1040165
                                                                            1040166
11100 9600 8400 7300 6400 5400 4600 3700 3000 2300 1800 1400                1040167
                                                                            1040168
1160019000 8800 7700 7000 5800 4900 4000 3200 2500 2000 1600                1040169
                                                                            1040170
1210010500 9200 8100 7500 6300 5200 4300 3400 2700 2200 1800                1040171
                                                                            1040172
                                                                            1040173
                                                                            1040174
                                                                            1040175
                                                                            1040176
                                                                            1040177
                                                                            1040178
                                                                            1040179
                                                                            1040180
                                                                            1040181
                                                                            1040182
                                                                            1040183
                                                                            1040184
C
C  ALTITUDE, MACHMAX (PLACARD LIMIT)
    0 50001000015000200002500030000340006500O                               1040185
113  125  138  153  168  182  191  200  200                                 1040186
C
C  MACH, CGLIFT, ALPHA
   .0     .5     .6     .7     .8     .9    .95   1.0    1.1    1.2    1.3    1.4  1040187
  1.5    1.6    1.7    1.8    1.9    2.0    2.1    2.2    2.3    2.4                1040188
-  .4     .0     .1     .2     .3     .4     .5     .6     .7     .8     .9    1.0  1040189
  1.1                                                                             1040190
- 9.0 -   .2    2.0    3.8    5.6    7.4    9.2   11.0   13.0   15.3   17.9        1040191
                                                                                  1040192
- 9.0 -   .2    2.0    3.8    5.6    7.4    9.2   11.0   13.0   15.3   17.9        1040193
                                                                                  1040194
- 9.0 -   .2    2.0    3.7    5.4    7.2    9.0   10.8   12.9   15.3   18.0        1040195
                                                                                  1040196
- 8.2 -   .2    1.8    3.5    5.0    6.7    8.5   10.3   12.6   15.2   17.1        1040197
                                                                                  1040198
- 7.4 -   .2    1.6    3.0    4.4    5.8    7.4    9.3   11.6   15.0   19.5        1040199
                                                                                  1040200
- 6.6 -   .2    1.4    2.8    4.0    5.2    6.4    7.8    9.6   11.9   14.7   18.0  1040201
 21.8                                                                             1040202
- 6.5 -   .1    1.5    2.7    4.0    5.1    6.3    7.8    9.2   10.7   12.0   13.1  1040203
 14.0                                                                             1040204
- 5.8     .2    1.7    2.8    4.1    5.4    6.7    8.0    9.4   10.9   12.5   14.2  1040205
 16.0                                                                             1040206
- 4.4     .8    2.1    3.6    4.9    6.3    7.7    9.2   10.7   12.5   14.3   16.1  1040207
                                                                                  1040208
- 4.9     .3    1.6    3.1    4.8    6.3    7.8    9.3   10.8   12.7   14.5   16.2  1040209
                                                                                  1040210
- 6.9 -   .5    1.1    2.7    4.2    6.0    7.6    9.2   10.9   12.8   14.7   16.6  1040211
                                                                                  1040212
- 7.0 -   .6    1.0    2.8    4.5    6.2    8.0    9.7   11.6   13.5   15.5   17.6  1040213
                                                                                  1040214
- 7.8 -   .6    1.2    3.0    4.8    6.6    8.5   10.4   12.4   14.5   16.6        1040215
                                                                                  1040216
- 8.6 -   .6    1.4    3.3    5.2    7.2    9.2   11.2   13.3   15.6   17.8        1040217
```

Continued

Fig. 35 (continued)

- 8.5 -	.5	1.5	3.6	5.7	7.8	9.9	12.1	14.4	16.7
- 9.2 -	.4	1.8	4.0	6.2	8.5	10.7	13.0	15.4	17.8
- 9.5 -	.3	2.0	4.3	6.8	9.1	11.6	13.9	16.6	19.7
- 9.8 -	.2	2.2	4.7	7.3	9.8	12.5	14.9	17.8	21.2
-10.0	.0	2.5	5.0	7.5	10.5	13.4	16.0	19.0	22.4
-10.7	.1	2.8	5.4	8.4	11.3	14.4	17.0	19.1	
-11.0	.2	3.0	5.8	9.0	12.0	15.4	18.1	20.1	
-11.6	.4	3.4	6.2	9.6	12.8	16.4	19.2	21.2	

1040218
1040219
1040220
1040221
1040222
1040223
1040224
1040225
1040226
1040227
1040228
1040229
1040230
1040231
1040232
1040233
1040234
0253 CARDS

Fig. 35 (continued)

Appendix G

PROGRAM SUBROUTINES

The subroutines in Table 4, which constitute the main body of TACTICS, must always be used when running the program; all other subroutines are optional.

Table 4

TACTICS SUBROUTINES

Subroutine	Function
AERØDN	Computes aerodynamic variables and outputs change in velocity
AIM[a]	Aims fighter to obtain initial-condition values for V(I,5) and V(I,6) angles
ATMØS	Computes model atmosphere
ATTITUD	Computes attitude angles for the vehicles
AUXCØM	Calls output at specified times
CØØRD	Makes rectangular and spherical coordinate transformations
CRØSS	Computes vector cross product
DAUX	Computes derivatives for position and velocity
DECRD	Reads initial-condition data
DØT	Computes vector dot product
GEØFRC[a]	Computes force for geocentric integration
GEØCEN[a]	Determines earth rotation rate and finds unit geometric vectors
INCØND	Reads, computes, and prints initial conditions
INITS	Initializes conditions for integration
INTGRT	Calls integration and computes new velocities and ranges
LAG	Imposes time lag (see Section V)

[a]If storage is a problem, dummy decks may be substituted for these subroutines, since they are only used when special options are selected. (Since the routines in the main body refer to the special options, however, some type of deck must always be used to represent them.)

Table 4 (continued)

Subroutine	Function
LIMIT	Imposes aerodynamic and structural (lateral) acceleration constraints when applicable (see Section V)
MAG	Computes magnitude of a vector
MAIN	Orders operations
ØUTPUT	Prints output
PLACRD	Reduces throttle setting if there is a danger of exceeding maximum Mach number (placard limit)
RATES	Computes angular rates
STØRE[a]	Stores position and velocity when using restore option
TABINT	Interpolates aerodynamic tables
TABLER	Reads aerodynamic tables if used
THRST	Computes thrust of velocities
TRPØCN[a]	Makes local and inertial coordinate transformations
WEIGH	Computes weight of vehicles

[a]If storage is a problem, dummy decks may be substituted for these subroutines, since they are only used when special options are selected. (Since the routines in the main body refer to the special options, however, some type of deck must always be used to represent them.)

Appendix H

FORTRAN FLAGS

As is usually the case with large computer programs, TACTICS uses many FORTRAN flags for internal control and operation. Although in most cases the user need not be concerned with their definition or functional purpose, certain key flags require explanation. Some of those flags may be set in PØLICY or by an initial-condition data value to select a program option; others are important because they control or are indicators of various phases of program operation. The following is an explanatory list of those FORTRAN flags considered to be of primary importance:

IMISS: Whether program is to continue after finding closest missile approach.

IMISS = 0, the program stops after finding miss distance.

IMISS = 1, the program continues after finding missile miss; flag is set in PØLICY.

IMISS = 2, the program has determined the miss distance and is ready to continue; flag is set in AUXCØM, and is normally used in PØLICY as a criterion.

ISTØRE: Whether position and velocity values are to be restored to those existing at launch time.

ISTØRE = 0, the positions and velocities are not to be stored.*

ISTØRE = 1, the store option is to be used. The values of position and velocity are stored at launch, and the program returns to these values after computing the closest point of approach; flag is set in PØLICY and used in MAIN to trigger STØRE routine.

*ISTØRE is set at zero in STØRE routine to indicate that values have been restored, and is normally used in PØLICY as a criterion.

JPØL, KPØL, LPØL, MPØL, NPØL: To be used in PØLICY; initially set equal to 1 in INCØND.

JINTEG: The type of integration to be used; set by DATA 122. If integration is to be changed during a run, JINTEG may be reset in PØLICY.

JINTEG = 0, variable-step Adams-Moulton.

JINTEG = 1, fixed-step Runge-Kutta.

JINTEG = 2, fixed-step Adams-Moulton.

JINTEG = 3, variable-step Adams-Moulton with exact printout.

ISTART: Used within program to determine specific events.

ISTART = 0, the start of the program (time = 0.0).

ISTART = 1, value is set after the first integration step, and remains at this value until closest missile approach or until the end of the problem run; flag is set in MAIN.

ISTART = 2, the program is backing up to compute the closest missile approach; flag is set in INTGRT. If the program is to continue after computing miss distance, ISTART is automatically reset to 1 in AUXCØM.

ILAUN: Used within program to indicate status of missile launching.

ILAUN = 1, the missile has not been launched; flag is set initially in INCØND and is reset in CAPFLT if recall or restore options are used.

ILAUN = 2, launch criteria have been overshot because of an excessive step size, and program is to back up to approach launch on a smaller fixed step; flag is set in LAUNCH (see Section XI).

ILAUN = 3, the missile has been launched; flag is set in LAUNCH and is usually used as a criterion in PØLICY.

ILAUN = 4, the program has backed up to the step before launch and will approach on fixed step size integration; flag is set in INITS. This is to keep CAPFLT from resetting ILAUN to 1, which would prevent the missile derivatives from being integrated.

JATMØS: Whether the velocities of the fighter and target are to be read in ft/sec or Mach number for initial data.

JATMØS = 0 (ft/sec)

JATMØS = 1 (Mach number)

KINTEG: Whether a flat or spherical earth gravitational force representation is to be used (see Section IV and Appendix B).

KINTEG = 0, the flat-earth option is to be used.

KINTEG = 1, the round-earth option is to be used.

ICAP: This flag may serve several different functions, all related to the flight status of vehicle 2. Its primary purpose is to avoid needless computations and integration of the equations of motion for vehicle 2 when it is not being used, i.e., captive flight or undefined motion. The ICAP value should be read in as 0, 1, 2, or 3 (DATA 28).

ICAP = 0, computations and integration will occur for vehicle 2. If a nonzero value for V(2,4) (DATA 33) has been entered as an initial-condition value, the ILAUN = 3 flight status is assumed to exist. If DATA 33 is an exact zero value, the ILAUN = 1 status is assumed and computation will occur, presumably in anticipation of the ground launch of vehicle 2 at some subsequent time. If an intercept problem is involved, the target vehicle will be vehicle 3 for miss-distance computation.

ICAP = 1, vehicle 2 is in captive flight on vehicle 1; if launch occurs, it will intercept against

vehicle 3. No computations for vehicle 2 are performed prior to reaching criteria for launch.

ICAP = 2, vehicle 2 is not being used; it will automatically be located at the zero origin with zero velocity and acceleration. No computations or integration will occur for vehicle 2.

ICAP = 3, vehicle 2 is in captive flight on vehicle 3; if launch occurs, it will intercept against vehicle 1. No computations for vehicle 2 are performed prior to reaching criteria for launch.

Appendix I

MODEL ATMOSPHERE

A model atmosphere may be postulated in either tabular or analytic form depending on the degree of realism required by the problem. For most intercept problems, an exponential analytic form is sufficient, and no provision is made in TACTICS for incorporating tabular values. The air density ρ, pressure p, and absolute temperature T may be approximated as a function of altitude z from the following expressions, which are consistent with those given for the ICAO standard atmosphere (although constants are not represented to the same degree of significance).[4]

For the troposphere ($0 < z < z_s$ (ft))

$$t = 59 - 0.00357\, z \ (°F)$$

$$p = p_a \left(1 - \frac{0.00357\, z}{518.4}\right)^{5.256} \tag{140}$$

$$\rho = 0.002378 \left(1 - \frac{0.00357\, z}{518.4}\right)^{4.256} \quad \text{slug/ft}^3$$

For the stratosphere ($z_s < z < 86{,}000$ ft)

$$t = -67°F$$

$$p = 489.456\, e^{-(z-z_s)/h_s} \quad \text{lb/ft}^2$$

$$h_s = h_a \frac{T_s}{T_a} \quad \text{ft} \tag{141}$$

$$\rho = \frac{p}{gh_s} \quad \text{slug/ft}^3$$

where

p = pressure at z ft (lb/ft^2)

T = absolute temperature (^oR) = t + 459.4 (^oF)

t = temperature at altitude z ft ($z < z_s$) (^oF)

ρ = air density at z ft

h_a = pressure head at sea level (ft)

p_a = pressure at sea level (lb/ft^2)

T_a = absolute temperature at sea level (^oR)

h_s = pressure head at z_s (ft)

p_s = pressure at z_s (calculated from Eqs. (141)) (lb/ft^2)

T_s = absolute temperature at altitude z_s (^oR)

z_s = an altitude defining the beginning of the stratosphere and isothermal conditions (ft)

The above relationships are applicable for the so-called standard atmosphere, with a temperature gradient of $-3.57^oF/1000$ ft in the troposphere. Assuming the atmosphere can be characterized by isothermal conditions above an altitude of 35,300 ft at a temperature of -67^oF yields the following:

h_a = 27,650 ft	h_s = 20,916 ft
p_a = 2,116 lb/ft^2	p_s = 489.456 lb/ft^2
T_a = 518.4 ^oR	T_s = 392.4^oR
t_a = 59^oF	t_s = -67^oF
	z_s = 35,300 ft

The speed of sound, s, is equal to $\sqrt{1.4\ p/\rho}$ ft/sec. Mach number is related to vehicle speed V by $M = V/s$.

REFERENCES

1. Boehm, B., *ROCKET: RAND's Omnibus Calculator of the Kinematics of Earth Trajectories*, The Rand Corporation, RM-3534-PR, July 1963.
2. Causey, R., and W. L. Frank, *Adams-Moulton, Runge-Kutta Integration Subroutine*, Space Technology Laboratories, SHARE Distribution Agency, Report 602, November 1958.
3. Hildebrand, F. B., *Introduction to Numerical Analysis*, McGraw-Hill Book Company, Inc., New York, 1956.
4. Minzer, R. A., W. S. Ripley, and T. P. Condron, *U.S. Extension to the ICAO Standard Atmosphere*, U.S. Department of Commerce, U.S. Government Printing Office, Washington, D.C., 1958.

TACTICS PROGRAM - INPUT FORM[a]

VEHICLE TABLE FLAGS (0 OR 1)[b]

6	12	18
#1	#2	#3

A

TITLE CARD[b]

Columns 5 10 15 20 25 30 35 40 45 50 55 60 65 70 72

MAIN SET OF INITIAL CONDITIONS

Vehicle #1

1–4	6–18	20–32	34–46	48–60	62–74
0,0,0,1	Cartesian (0)	x	y	z	Initial weight
0,0,0,6	Spherical (1)	r	θ	ϕ	Reference area
0,0,1,1	Cartesian (0)	$\dot{x}$	$\dot{y}$	$\dot{z}$	a_{Smax} (lateral g's)
0,0,1,6	Spherical (1 or 2)	V	θ_V	γ	Mach no. flag

Vehicle #2

1–4	6–18	20–32	34–46	48–60	62–74
0,0,2,5	Initial weight	Reference area	a_{Smax} (lateral g's)	Captive flight (0,1,2,3)	Maximum range

Vehicle #3

1–4	6–18	20–32	34–46	48–60	62–74
0,0,4,0	Cartesian (0)	x	y	z	Initial weight
0,0,4,5	Spherical (1)	r	θ	ϕ	Reference area
0,0,5,0	Cartesian (0)	$\dot{x}$	$\dot{y}$	$\dot{z}$	a_{Smax} (lateral g's)
0,0,5,5	Spherical (1 or 2)	V	θ_V	γ	

PRINT

1–4	6–18	20–32	34–46	48–60	62–74
0,0,6,2	Initial time	Print interval	Termination time		

GUIDANCE (3 VEHICLES)

1–4	Number of time constants	$\tau(1)$	$\tau(2)$	$\tau(3)$	λ navigation constant
0,0,6,5					
0,0,7,0					
0,0,7,5					

AERODYNAMICS AND PROPULSION (3 VEHICLES)

1–4	C_{Lmax}	C_{D0}	dC_D/dC_L^2	$dC_L/d\alpha$	a_0

0045	Spherical (1)	r	θ	ϕ	Reference area
0050	Cartesian (0)	$\dot{x}$	$\dot{y}$	$\dot{z}$	$a_{S_{max}}$ (lateral g's)
0055	Spherical (1 or 2)	V	θ_V	γ	

PRINT

0062	Initial time	Print interval	Termination time		

GUIDANCE (3 VEHICLES)

	Number of time constants	$\tau(1)$	$\tau(2)$	$\tau(3)$	λ navigation constant
0065					
0070					
0075					

AERODYNAMICS AND PROPULSION (3 VEHICLES)

	$C_{L_{max}}$	C_{D0}	dC_D/dC_L^2	$dC_L/d\alpha$	α_0
0080					
0085					
0090					

	Military thrust	Afterburner thrust	Specific impulse	Burnout weight	Boost acceleration
0095					
0100					
0105					

ROUND-EARTH OPTION[c]

0110	Round-earth flag	Altitude (#1)	Longitude (#1)	Latitude (#1)	Velocity flag (#1)
0115	Rotation (1)	Altitude (#3)	Longitude (#3)	Latitude (#3)	Velocity flag (#3)
0120	Origin longitude	Origin latitude			

MISCELLANEOUS OPTIONS AND EXTRA INPUTS[c]

0122	Integration flag	No. of significant digits req.	Miss calculation R_{min}	θ_V aiming error (#1)	θ_V aiming error (#2)
0127	θ_V aiming error (#3)	γ aiming error (#1)	γ aiming error (#2)	γ aiming error (#3)	Ballistic coefficient (#1)
0132	Ballistic coefficient (#2)	Ballistic coefficient (#3)			
d -					

[a]The following units are used: distance (ft), time (sec), velocity (ft/sec or Mach no.), acceleration (g's), angles (deg), weight (lb), area (ft^2).

[b]This card required.

[c]Leave this section blank if option not used.

[d]Last data card must have a minus sign in column 1.

DOCUMENT CONTROL DATA

1. ORIGINATING ACTIVITY THE RAND CORPORATION	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP
3. REPORT TITLE TACTICS: A THREE-BODY, THREE-DIMENSIONAL INTERCEPT SIMULATION PROGRAM	
4. AUTHOR(S) (Last name, first name, initial) Hutcheson, J. H. and R. L. Segerblom	

5. REPORT DATE	6a. TOTAL No. OF PAGES	6b. No. OF REFS.
October 1969	188	4
7. CONTRACT OR GRANT No. F44620-67-C-0045	8. ORIGINATOR'S REPORT No. RM-5759-PR	
9a. AVAILABILITY/LIMITATION NOTICES DDC-1	9b. SPONSORING AGENCY United States Air Force Project RAND	

10. ABSTRACT

TACTICS is a computer program written in FORTRAN IV which mathematically simulates the dynamics of flight in three-dimensional space of as many as three vehicles simultaneously. The purpose of the Memorandum is to acquaint users with the capabilities and basic theory of the program and to serve as a reference manual for those who wish to use the program. The first part of the Memorandum describes the theory of operation and is oriented toward those with a mathematical or technical background. The second part is concerned with how to use the program, i.e., how to provide input data, select options, and develop a flight program. A number of illustrative examples of a wide variety of problems are given in detail, including data and FORTRAN listings. In many cases it should be possible to set up a specialized problem by modifying or combining various features of the examples.

11. KEY WORDS

TACTICS (Air Combat Simulation)
Computer simulation
Tactical aircraft
Missiles
Interception
Air combat

DOCUMENT CONTROL DATA

1. ORIGINATING ACTIVITY THE RAND CORPORATION	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP

3. REPORT TITLE

TACTICS: A THREE-BODY, THREE-DIMENSIONAL INTERCEPT SIMULATION PROGRAM

4. AUTHOR(S) (Last name, first name, initial)

Hutcheson, J. H. and R. L. Segerblom

5. REPORT DATE	6a. TOTAL No. OF PAGES	6b. No. OF REFS.
October 1969	188	4

7. CONTRACT OR GRANT No.	8. ORIGINATOR'S REPORT No.
F44620-67-C-0045	RM-5759-PR

9a. AVAILABILITY/LIMITATION NOTICES	9b. SPONSORING AGENCY
DDC-1	United States Air Force Project RAND

10. ABSTRACT

TACTICS is a computer program written in FORTRAN IV which mathematically simulates the dynamics of flight in three-dimensional space of as many as three vehicles simultaneously. The purpose of the Memorandum is to acquaint users with the capabilities and basic theory of the program and to serve as a reference manual for those who wish to use the program. The first part of the Memorandum describes the theory of operation and is oriented toward those with a mathematical or technical background. The second part is concerned with how to use the program, i.e., how to provide input data, select options, and develop a flight program. A number of illustrative examples of a wide variety of problems are given in detail, including data and FORTRAN listings. In many cases it should be possible to set up a specialized problem by modifying or combining various features of the examples.

11. KEY WORDS

TACTICS (Air Combat Simulation)
Computer simulation
Tactical aircraft
Missiles
Interception
Air combat